```
     (FILE 'HOME' ENTERED AT 11:12:36 ON 20 MAY 2003)

     FILE 'USPAT2, EUROPATFULL, JAPIO, INPADOC, INSPEC, NLDB, PATOSEP,
     PATOSWO' ENTERED AT 11:13:01 ON 20 MAY 2003
L1           13 S (OPERATING(W)SYSTEM)(S)(EMULAT?(2A)OBJECT)
L2            5 S (EMULATION(W)OBJECT)
L3            0 S L2 AND OPERATING(W)SYSTEM
L4          143 S (EMULAT?(S)OBJECT)(P)(OPERATING(W)SYSTEM)
L5           43 S (EMULAT?(S)OBJECT)(S)(DIRECT? OR LINK? OR INTERFAC?)(S)(OPERA
L6           68 S (EMULAT?(3W)OBJECT)
L7           46 S (EMULAT?(2W)OBJECT)
L8           16 S (EMULAT?(3W)OBJECT)(P)(OPERATING(W)SYSTEM)

=>
```

```
Trying 3106016892...Open

Welcome to STN International!  Enter x:x
LOGINID:ssspta2700akr
PASSWORD:
TERMINAL (ENTER 1, 2, 3, OR ?):2


* * * * * * * * * *   Welcome to STN International   * * * * * * * * * *

 NEWS  1  Feb  2  Web Page URLs for STN Seminar Schedule - N. America
 NEWS  2  Dec 17  Expanded CAplus Coverage of US, Japanese, WIPO,
                  EPO, and German patents
 NEWS  3  Jan 18  ESBIOBASE - NEW FREE DISPLAY FORMATS, TRIAL
                  FORMAT ENHANCED
 NEWS  4  Feb  1  Addition of Machine-Translated Abstracts to CAplus
 NEWS  5  Feb  2  STEREO BOND SEARCH PROBLEM FIXED WITH STN EXPRESS 5.0C
 NEWS  6  Feb 14  Homology Searching for Nucleotide Sequences in DGENE
                  now available!
 NEWS  7  Feb 16  BIOTECHNOBASE NOW ON STN
 NEWS  8  Feb 22  New Database Producer Clusters Now Available on STN
 NEWS  9  Feb 28  Structure Search Limits Increased in REGISTRY,
                  ZREGISTRY, and CASREACT
 NEWS 10  Feb 28  Patent Information Now Searchable in CAOLD
 NEWS 11  Mar  1  New IMSDIRECTORY Provides Pharma Company Details

 NEWS EXPRESS   FREE UPGRADE 5.0C NOW AVAILABLE
 NEWS HOURS     STN Operating Hours Plus Help Desk Availability
 NEWS INTER     General Internet Information
 NEWS LOGIN     Welcome Banner and News Items
 NEWS PHONE     Direct Dial and Telecommunication Network Access to STN
 NEWS WWW       CAS World Wide Web Site (general information)

Enter NEWS followed by the item number or name to see news on that
specific topic.

  All use of STN is subject to the provisions of the STN Customer
  agreement.  Please note that this agreement limits use to scientific
  research.  Use for software development or design or implementation
  of commercial gateways or other similar uses is prohibited and may
  result in loss of user privileges and other penalties.

* * * * * * * * * * * * * * STN  Columbus  * * * * * * * * * * * * * * *

FILE 'HOME' ENTERED AT 09:59:46 ON 06 MAR 2000

=> file europatfull inspec japio nldb patosep patoswo promt uspatfull

COST IN U.S. DOLLARS                              SINCE FILE       TOTAL
                                                       ENTRY     SESSION
FULL ESTIMATED COST                                     0.45        0.45

FILE 'EUROPATFULL' ENTERED AT 10:01:16 ON 06 MAR 2000
COPYRIGHT (c) 2000 WILA Verlag Muenchen (WILA)

FILE 'INSPEC' ENTERED AT 10:01:16 ON 06 MAR 2000
Compiled and produced by the IEE in association with FIZ KARLSRUHE
COPYRIGHT 2000 (c) INSTITUTION OF ELECTRICAL ENGINEERS (IEE)
```

```
FILE 'JAPIO' ENTERED AT 10:01:16 ON 06 MAR 2000
COPYRIGHT (C) 2000 Japanese Patent Office (JPO)

FILE 'NLDB' ENTERED AT 10:01:16 ON 06 MAR 2000
COPYRIGHT (C) 2000 Information Access Company.  All rights reserved.

FILE 'PATOSEP' ENTERED AT 10:01:16 ON 06 MAR 2000
COPYRIGHT (c) 2000 WILA Verlag Muenchen (WILA)

FILE 'PATOSWO' ENTERED AT 10:01:16 ON 06 MAR 2000
COPYRIGHT (c) 2000 WILA Verlag Muenchen (WILA)

FILE 'PROMT' ENTERED AT 10:01:16 ON 06 MAR 2000
COPYRIGHT (C) 2000 Information Access Company.  All rights reserved.

FILE 'USPATFULL' ENTERED AT 10:01:16 ON 06 MAR 2000
CA INDEXING COPYRIGHT (C) 2000 AMERICAN CHEMICAL SOCIETY (ACS)

=> s (pos or (point(w)of(w)sale) or register)

   3 FILES SEARCHED...
L1       384825 (POS OR (POINT(W) OF(W) SALE) OR REGISTER)

=> s l1 and (emulat and (application(3w)develop?))

AND IS NOT A RECOGNIZED COMMAND
The previous command name entered was not recognized by the system.
For a list of commands available to you in the current file, enter
"HELP COMMANDS" at an arrow prompt (=>).

=> s l1 and emulat? and (application(3w)develop?)

   2 FILES SEARCHED...
   7 FILES SEARCHED...
L2          273 L1 AND EMULAT? AND (APPLICATION(3W) DEVELOP?)

=> s l1 and (emulat?(s)(application(3w)develop?))

   2 FILES SEARCHED...
PROXIMITY OPERATOR LEVEL NOT CONSISTENT WITH
FIELD CODE - 'AND' OPERATOR ASSUMED 'EMULAT?(S)(APPLICATI'
   7 FILES SEARCHED...
L3           43 L1 AND (EMULAT?(S)(APPLICATION(3W) DEVELOP?))

=> s l3 and (emulat?(2w)module)

L4            1 L3 AND (EMULAT?(2W) MODULE)

=> d


L4   ANSWER 1 OF 1  USPATFULL
AN     1998:59535  USPATFULL
TI     Computer emulator
IN     Ogata, Hideaki, Nagano-Ken, Japan
       Tanimoto, Akihito, Nagano-Ken, Japan
       Nakaoka, Yasushi, Nagano-Ken, Japan
       Kojima, Masanori, Nagano-Ken, Japan
       Akahori, Yutaka, Nagano-Ken, Japan
PA     Seiko Epson Corporation, Tokyo, Japan (non-U.S. corporation)
PI     US 5758124  19980526
AI     US 1995-559223  19951117 (8)
PRAI   JP 1994-309813      19941118
DT     Utility
LN.CNT 883
```

```
INCL    INCLM: 395/5[illegible]000
        INCLS: 395/5[illegible]00; 395/421.030; 364/DIG.001; 364/232.300; 364/DIG.002;
               364/927.810; 364/955.500
NCL     NCLM:  395/500.480
        NCLS:  395/527.000; 711/213.000
IC      [6]
        ICM: G06F009-455
EXF     395/500; 395/375; 395/750; 395/800; 395/416; 395/527; 395/421.03;
        364/200MSFile; 364/900MSFile; 364/927.81


=> d kwic


L4   ANSWER 1 OF 1  USPATFULL
AB      . . .  The emulator emulates routines that are called via a jump
table
        such as the BIOS. Control is transferred to an emulation
      module not by directly trapping the procedure to call the BIOS
        but by placing a privileged instruction (a halt instruction, for.  . .
        trap through the execution of the privileged instruction. An identifier
        is placed after the halt instruction and a needed BIOS emulation
      module is called by a dispatcher using this identifier.
        Therefore, normal operation can be obtained even if there is a
resident.
        . .
SUMM    A conventional virtual computer emulator of this type which
        makes application programs developed for a target
        machine executable on an executing machine by individually
      emulating OS function calls, BIOS function calls, I/O
        instructions and interrupt tables of the target machine has been known
        as disclosed.  . .
DETD    . . .  also used when one execution module calls another module and
        when modules that are to communicate with each other mutually
      register the other's entry point EP for direct calling. For
        example, when one execution module calls another execution module, the
        service.  . .
DETD    Subsequent to the routine to register interrupts, it is
        determined whether the calling module is a module which has entry point
        EP for direct calling in.  . .
DETD    . . .  module for emulation of the BIOS will be explained below. FIG.
        14 is a flow chart showing the routine to register a BIOS
        emulation function. Prior to the onset of this routine, the BIOS for a
        PC-9800 is allocated to the.  . .
DETD    . . .  interrupt routine for said interrupt and the BIOS is called.
        The contents of jump table JT corresponding to the BIOS emulator
      module currently being established, i.e., the go-to address, are
        then obtained and saved in a prescribed area. This address is saved.  .
        .  request is made using the same software interrupt, the process to be
        executed varies depending on the value in the register, and
        when a routine described below in which BIOS emulation is not performed
        is requested, control may be transferred to.  . .
DETD    With regard to various execution modules including BIOS
      emulation modules, module name (characters) TM and
        call address A at the minimum need to be registered in dispatcher table
        DT at the.  . .
DETD    . . .  into an address in which a HLT instruction and an identifier
        are embedded in the initialization process when this BIOS
      emulation module was registered (see FIG. 14), in
        actuality, the HLT instruction, not the BIOS program, is executed (E2).
DETD    . . .  following the HLT instruction, i.e., the identifier. After
        obtaining the identifier, kernel KR requests dispatcher DP to call the
        BIOS emulation module using said identifier (E6).
        Receiving this request, dispatcher DP refers to dispatcher table DT
        registered and calls the BIOS routine prepared as an emulation
```

```
     module (E7).
DETD   If the BIOS routine prepared as an emulation module
       is a routine to access the floppy disk device, for example, it controls
       compound I/O ports 54 and floppy disk.  .  .  as DOS/V hardware. FIG.
17

       shows an example in which data is read from floppy disk device 82. The
       BIOS emulation module reads 256 bytes of data from
       floppy disk device 82 and saves it in buffer area FBF (F1 and F2). The
       data read into buffer area FBF is transferred to disk buffer FB of the
       system by the BIOS emulation module.
DETD   After these sequences, processing by the BIOS emulation
     module is completed and control returns to the next instruction
       after the BIOS call of application program APP (E8). The return.  .  .

DETD   .  .  .  instruction (a HLT instruction in the embodiment) is executed
       in the called routine, whereupon control is transferred to the BIOS
     emulation module. Therefore, the BIOS
     emulation module can be activated through reliable
       detection of the call not only where the BIOS is called via a software
       interrupt.  .  .  called by obtaining its entry point through referring
       to the jump table. Moreover, both the resident program and the BIOS
     emulation module can be normally operated even where
       there is an incorporated resident program of the type that operates
       after taking over.  .  .
DETD   .  .  .  large program to emulate a large number of functions may be
       efficiently developed. Moreover, since execution modules, including a
       BIOS emulation module, are modulized in a manner
       similar to the functioning of hardware, if application program APP is
       one that directly accesses.  .  .
DETD   Subsequently, when the BIOS is called by application program APP,
kernel
       KR activates immediately and transfers control to the BIOS
     emulation module via dispatcher DP. Therefore, the
       BIOS call is reliably detected in the second embodiment as well as in
       the first.  .  .


=> d ab


L4   ANSWER 1 OF 1  USPATFULL
AB     What is disclosed is an emulator that emulates on an execution machine
       the operation of a target machine. The emulator emulates routines that
       are called via a jump table such as the BIOS. Control is transferred to
       an emulation module not by directly trapping the
       procedure to call the BIOS but by placing a privileged instruction (a
       halt instruction, for example) in the area called and by causing a trap
       through the execution of the privileged instruction. An identifier is
       placed after the halt instruction and a needed BIOS emulation
     module is called by a dispatcher using this identifier.
       Therefore, normal operation can be obtained even if there is a resident
       program which rewrites the jump table.


=> d


L4   ANSWER 1 OF 1  USPATFULL
AN     1998:59535  USPATFULL
TI     Computer emulator
IN     Ogata, Hideaki, Nagano-Ken, Japan
       Tanimoto, Akihito, Nagano-Ken, Japan
       Nakaoka, Yasushi, Nagano-Ken, Japan
       Kojima, Masanori, Nagano-Ken, Japan
       Akahori, Yutaka, Nagano-Ken, Japan
```

```
PA      Seiko Epson Corporation, Tokyo, Japan (non-U.S. corporation)
PI      US 5758124  19980526
AI      US 1995-559223  19951117 (8)
PRAI    JP 1994-309813      19941118
DT      Utility
LN.CNT  883
INCL    INCLM: 395/500.000
        INCLS: 395/527.000; 395/421.030; 364/DIG.001; 364/232.300; 364/DIG.002;
               364/927.810; 364/955.500
NCL     NCLM:  395/500.480
        NCLS:  395/527.000; 711/213.000
IC      [6]
        ICM: G06F009-455
EXF     395/500; 395/375; 395/750; 395/800; 395/416; 395/527; 395/421.03;
        364/200MSFile; 364/900MSFile; 364/927.81


=> d hist


     (FILE 'HOME' ENTERED AT 09:59:46 ON 06 MAR 2000)

     FILE 'EUROPATFULL, INSPEC, JAPIO, NLDB, PATOSEP, PATOSWO, PROMT,
     USPATFULL' ENTERED AT 10:01:16 ON 06 MAR 2000
L1        384825 S (POS OR (POINT(W)OF(W)SALE) OR REGISTER)
L2           273 S L1 AND EMULAT? AND (APPLICATION(3W)DEVELOP?)
L3            43 S L1 AND (EMULAT?(S)(APPLICATION(3W)DEVELOP?))
L4             1 S L3 AND (EMULAT?(2W)MODULE)

=> s l3 and execut?

L5           37 L3 AND EXECUT?

=> s l3 and (application(s)execut?)

PROXIMITY OPERATOR LEVEL NOT CONSISTENT WITH
FIELD CODE - 'AND' OPERATOR ASSUMED 'PPLICATION(S)EXECUT?'
L6           32 L3 AND (APPLICATION(S) EXECUT?)

=> d 1-10


L6      ANSWER 1 OF 32  EUROPATFULL  COPYRIGHT  2000 WILA

PATENT APPLICATION - PATENTANMELDUNG - DEMANDE DE BREVET

AN      961193  EUROPATFULL  ED 19991212  EW 199948  FS OS
TIEN    Secure computing device.
TIDE    Sichere Rechnervorrichtung.
TIFR    Dispositif ordinateur securise.
IN      Laczko Sr., Frank L., 301 S. Jupiter, P.O. Box 23, Allen, Collin
County,
        Texas, US
PA      Texas Instruments Incorporated, 7839 Churchill Way, Dallas, Texas
75251,
        US
SO      Wila-EPZ-1999-H48-T2a
DS      R AT; R BE; R CH; R CY; R DE; R DK; R ES; R FI; R FR; R GB; R GR; R IE;
        R IT; R LI; R LU; R MC; R NL; R PT; R SE; R AL; R LT; R LV; R MK; R RO;
        R SI
PIT     EPA2 EUROPAEISCHE PATENTANMELDUNG
PI      EP 961193             A2 19991201
OD                               19991201
AI      EP 1999-201705           19990528
PRAI    US 1998-87229            19980529
```

```
         US 1998-871          19980529
         US 1998-872          19980529
         US 1998-87230        19980529
'IC      ICM  G06F001-00
         ICS  G06F012-14

L6       ANSWER 2 OF 32  EUROPATFULL  COPYRIGHT  2000 WILA

PATENT APPLICATION - PATENTANMELDUNG - DEMANDE DE BREVET

AN       786722  EUROPATFULL  ED 19970810  EW 199731  FS OS
TIEN     A method and system for improving emulation performance.
TIDE     Verfahren und Anordnung zur Leistungsverbesserung von Emulation.
TIFR     Procede et systeme pour ameliorer la performance d'emulation.
IN       Kahle, James A., 5312 Tortuga Trail, Austin, Texas 78731, US;
         Mallick, Soummya, 13032 Partridge Bend Drive, Austin, Texas 78729, US
PA       INTERNATIONAL BUSINESS MACHINES CORPORATION, Armonk, NY 10504, US
SO       Wila-EPZ-1997-H31-T2a
DS       R DE; R FR; R GB
PIT      EPA1 EUROPAEISCHE PATENTANMELDUNG
PI       EP 786722           A1 19970730
OD                              19970730
AI       EP 1996-309517         19961227
PRAI     US 1996-581793         19960125
IC       ICM  G06F009-318

L6       ANSWER 3 OF 32  EUROPATFULL  COPYRIGHT  2000 WILA

PATENT APPLICATION - PATENTANMELDUNG - DEMANDE DE BREVET

AN       712077  EUROPATFULL  UP 19970408  EW 199620  FS OS  STA R
TIEN     Microcontroller with provisions for emulation.
TIDE     Mikrokontroller mit Einrichtung zur Emulation.
TIFR     Microcontroleur avec dispositions pour l'emulation.
IN       Iles, Alexander L., 6912 Chinook Drive, Austin, Texas 78736, US;
         Jelemensky, Joseph, 4606 Palisade Drive, Austin, Texas 78731, US;
         Yishay, Oded, 7790 Lekewood Drive, Austin, Texas 78750, US
PA       MOTOROLA, INC., 1303 East Algonquin Road, Schaumburg, IL 60196, US
SO       Wila-EPZ-1996-H20-T2a
DS       R DE; R FR; R GB
PIT      EPA1 EUROPAEISCHE PATENTANMELDUNG
PI       EP 712077           A1 19960515
OD                              19960515
AI       EP 1995-116864         19951026
PRAI     US 1994-333658         19941103
IC       ICM  G06F011-00

L6       ANSWER 4 OF 32  EUROPATFULL  COPYRIGHT  2000 WILA

PATENT APPLICATION - PATENTANMELDUNG - DEMANDE DE BREVET

AN       685793  EUROPATFULL  ED 19991107  EW 199549  FS OS  STA B
TIEN     Emulation device, system and method with distributed control of test
         interfaces in clock domains.
TIDE     Vorrichtung, System und Verfahren zur Emulation mit dezentraler
         Steuerung im Taktbereich-Pruefungsschnittsteller.
TIFR     Dispositif, systeme et procede d'emulation avec commande decentralisee
         d'interface de test.
IN       Swoboda, Gary L., 4435 Balboa, Sugar Land, TX 77479, US
PA       TEXAS INSTRUMENTS INCORPORATED, 13500 North Central Expressway, Dallas
         Texas 75265, US
SO       Wila-EPZ-1995-H49-T2a
DS       R DE; R FR; R GB; R IT; R NL
PIT      EPA2 EUROPAEISCHE PATENTANMELDUNG
PI       EP 685793           A2 19951206
```

```
OD                                  19951206
AI      EP 1995-30                  19950309
PRAI    US 1994-208543              19940309
        US 1994-209127              19940309
        US 1994-208469              19940309
IC      ICM  G06F011-26

L6      ANSWER 5 OF 32  EUROPATFULL  COPYRIGHT  2000 WILA

GRANTED PATENT - ERTEILTES PATENT - BREVET DELIVRE

AN      469507  EUROPATFULL  ED 19970307  EW 199637  FS PS
TIEN    Integrated circuit comprising a standard cell, an application cell and
a
        test cell.
TIDE    Integrierte Schaltung mit einer Standardzelle, einer Anwendungszelle
und
        einer Pruefzelle.
TIFR    Circuit integre comprenant une cellule standard, une cellule
        d'application et une cellule de test.
IN      Dartois, Luc, 98, avenue Paul Denis Huet, F-78955 Carrieres sous
Poissy,
        FR;
        Dulongpont, Jacques, 6, rue Albert Thomas, F-95300 Pontoise, FR;
        Reusens, Peter, Warande 121, B-9270 Laarne, BE
PA      ALCATEL MOBILE COMMUNICATION FRANCE, 15, rue de la Baume, 75008 Paris,
        FR
SO      Wila-EPS-1996-H37-T2
DS      R AT; R BE; R CH; R DE; R DK; R ES; R FR; R GB; R GR; R IT; R LI; R LU;
        R NL; R SE
PIT     EPB1 EUROPAEISCHE PATENTSCHRIFT
PI      EP 469507             B1 19960911
OD                               19920205
AI      EP 1991-112697           19910729
PRAI    FR 1990-9978             19900803
REP     EP 315475    A           EP 358376    A
        US 4698588   A
REN     COMPUTER DESIGN, vol. 29, no. 17, 1 septembre 1990, Littleton,
        Massachusetts, US, pages 94-112; J. GABAY: "How much can
Design-for-test
        reduce the need for testing?" IBM TECHNICAL DISCLOSURE BULLETIN, vol.
        25, no. 2, juillet 1982, NEW YORK, US, page 709; E.F. HAHN et al.:
"VLSI
        testing by on-chip error detection"
IC      ICM  G06F011-26
        ICS  G01R031-28

L6      ANSWER 6 OF 32  EUROPATFULL  COPYRIGHT  2000 WILA

GRANTED PATENT - ERTEILTES PATENT - BREVET DELIVRE

AN      437550  EUROPATFULL  ED 19971221  EW 199750  FS PS
TIEN    INFORMATION PROCESSING SYSTEM EMULATION APPARATUS AND METHOD.
TIDE    VERFAHREN UND VORRICHTUNG ZUR EMULATION EINES
        INFORMATIONSVERARBEITUNGSSYSTEMS.
TIFR    METHODE ET APPAREIL D'EMULATION POUR UN SYSTEME DE TRAITEMENT DE
        L'INFORMATION.
IN      MORSS, Stephen, 1 Fairmount Avenue, Somerville, MA 02144, US;
        DREYFUS, Boris, 77 Bow Street, Lexington, MA 02173, US
PA      WANG LABORATORIES, INC., 600 Technology Park Drive, Billerica, MA
01821,
        US
SO      Wila-EPS-1997-H50-T2
DS      R BE; R DE; R FR; R GB; R NL
PIT     EPB1 EUROPAEISCHE PATENTSCHRIFT   (Internationale Anmeldung)
```

```
PI      EP 437550           B1 19971210
OD                             19910724
AI      EP 1990-905836         19900328
PRAI    US 1989-367297         19890615
RLI     WO 90-US1642        900328 INTAKZ
        WO 9016027          901227 INTPNR
REP     EP 168034   A          EP 205949   A
        EP 260568   A          US 3938101  A
        US 4875186  A
IC      ICM  G06F013-10
        ICS  G06F009-44

L6      ANSWER 7 OF 32  EUROPATFULL  COPYRIGHT  2000 WILA

GRANTED PATENT - ERTEILTES PATENT - BREVET DELIVRE

AN      372834  EUROPATFULL  ED 19970108  EW 199625  FS PS
TIEN    Translation technique.
TIDE    Uebersetzungsverfahren.
TIFR    Methode de traduction.
IN      Goettelmann, John Charles, 1846 Southeast Drive, Point Pleasant New
        Jersey 08742, US;
        Hiller, Ronald George, 1 Crawford Place, Old Bridge New Jersey 08857,
        US;
        Krantzler, Irvan Jay, 17 Alameda Court, Eatontown New Jersey 07724, US;
        Macey, Christopher James, 54 Elm Place, Red Bank New Jersey 07701, US;
        Tuomenoksa, Mark Logan, 20 Frances Street, Shrewsbury New Jersey 07702,
        US
PA      AT&T Corp., 32 Avenue of the Americas, New York, NY 10013-2412, US
SO      Wila-EPS-1996-H25-T2
DS      R DE; R FR; R GB; R IT
PIT     EPB1 EUROPAEISCHE PATENTSCHRIFT
PI      EP 372834           B1 19960619
OD                             19900613
AI      EP 1989-312500         19891130
PRAI    US 1988-280767         19881206
REP     EP 168034   A
REN     SIGPLAN NOTICES vol. 22, no. 7, July 1987, YORKTON HEIGHTS N.Y. pages 1
        - 13; C. MAY: 'Mimic: A fast system/370 simulator'
IC      ICM  G06F009-455
        ICS  G06F009-45

L6    ANSWER 8 OF 32  PROMT  COPYRIGHT 2000 IAC

ACCESSION NUMBER:     97:253265  PROMT
TITLE:                Multimedia still drives the game
AUTHOR(S):            Ohr, Stephan
SOURCE:               Electronic Engineering Times, (5 May 1997) pp. 83.
                      ISSN: 0192-1541.
LANGUAGE:             English
WORD COUNT:             3470
                      *FULL TEXT IS AVAILABLE IN THE ALL FORMAT*

L6    ANSWER 9 OF 32  USPATFULL
AN      2000:25940  USPATFULL
TI      Processor condition sensing circuits, systems and methods
IN      Swoboda, Gary L., Sugar Land, TX, United States
        Ehlig, Peter N., Houston, TX, United States
PA      Texas Instruments Incorporated, Dallas, TX, United States (U.S.
        corporation)
PI      US 6032268  20000229
AI      US 1992-832661  19920204 (7)
RLI     Continuation of Ser. No. US 1989-388286, filed on 31 Jul 1989, now
        abandoned And a continuation-in-part of Ser. No. US 1987-93463, filed
on
```

```
          4 Sep 1987, now abandoned Ser. No. Ser. No.    1987-140055, filed on 31
          Dec 1987, now patented, Pat. No. US 5109494 Ser. No. Ser. No. US
          1987-140192, filed on 31 Dec 1987, now patented, Pat. No. US 5101498
          Ser. No. Ser. No. US 1989-347968, filed on 4 May 1989, now abandoned
          Ser. No. Ser. No. US 1989-347969, filed on 4 May 1989, now abandoned
          Ser. No. Ser. No. US 1989-347605, filed on 4 May 1989, now abandoned
          Ser. No. Ser. No. US 1989-347596, filed on 4 May 1989, now patented,
          Pat. No. US 5072418 Ser. No. Ser. No. US 1989-347615, filed on 4 May
          1989, now patented, Pat. No. US 5142677 Ser. No. Ser. No. US
          1989-347966, filed on 4 May 1989, now patented, Pat. No. US 5155812 And
          Ser. No. US 1989-347967, filed on 4 May 1989, now abandoned
DT        Utility
LN.CNT 3972
INCL      INCLM: 714/030.000
          INCLS: 714/034.000; 714/726.000
NCL       NCLM:  714/030.000
          NCLS:  714/034.000; 714/726.000
IC        [7]
          ICM: G01R031-3177
          ICS: G01R031-3185
EXF       371/16.2; 371/19; 371/22.3; 371/22.5; 714/30; 714/34; 714/726

L6     ANSWER 10 OF 32  USPATFULL
AN        1999:143711  USPATFULL
TI        Pin management of accelerator for interpretive environments
IN        Adams, Phillip M., Salt Lake City, UT, United States
PA        Novell, Inc., Provo, UT, United States (U.S. corporation)
PI        US 5983310  19991109
AI        US 1997-799192  19970213 (8)
DT        Utility
LN.CNT 1414
INCL      INCLM: 711/006.000
          INCLS: 711/163.000; 711/167.000; 711/123.000; 709/102.000; 709/103.000;
                 709/301.000; 710/008.000
NCL       NCLM:  711/006.000
          NCLS:  711/163.000; 711/167.000; 711/123.000; 709/102.000; 709/103.000;
                 709/301.000; 710/008.000
IC        [6]
          ICM: G06F009-32
EXF       711/6; 711/163; 711/167; 711/125; 711/123; 711/123-5; 395/681; 395/672;
          395/826; 395/712; 395/672-3; 709/102-4; 709/301; 710/6-8


=> d ab 1


L6        ANSWER 1 OF 32  EUROPATFULL  COPYRIGHT  2000 WILA
ABEN      A secure computing system (100) stores a program, preferably the real
          time operating system (210), that is encrypted with a private key. A
          boot ROM (135) on the same integrated circuit as the data processor and
          inaccessible from outside includes an initialization program and a
          public key corresponding to the private key. On initialization the boot
          ROM decrypts at least a verification portion of the program. On
          verification normal operation is enabled. On non-verification, the
          system could be disabled, or that application program could be
disabled.
          A diagnostic program is stored at predetermined non-relocatable
physical
          address in memory. The program is made non-relocatable using a special
          table look-aside buffer (137) having a fixed virtual address
        register (611) and a corresponding fixed physical address
        register (641). The secure computing system prevents
          unauthorized use of compressed video data stored in a
first-in-first-out
          memory buffer by encrypting the compressed video data stream using at
```

least a par f the chip identity number as encryption key (703). The data is recalled from memory (705) and decrypted (706) as needed for video decompression. The debugger/emulator tool commonly employed in program development is protected by a private encryption key used to encrypt at least verification token for the program. Upon each initialization of the debugger/ emulator, the secure computer system decrypts the verification token employing public decryption key (805) to indicate whether the program is secure or non-secure. <image>

=> d ab 2

L6 ANSWER 2 OF 32 EUROPATFULL COPYRIGHT 2000 WILA

ABEN A system and method for improving the performance of a processor that emulates a guest instruction where the guest instruction includes a first and second operand. The first operand is stored in a general purpose **register**, and the second operand is stored in a special-purpose **register**. The method and system provides a host instruction that performs an operation using the first operand and the second operand without moving the second operand from the special-purpose **register** into the general purpose **register**. This reduces the number of instructions in the semantic routines necessary to operate on immediate data from guest instructions and increases emulation performance. <image>

=> d ab 3

L6 ANSWER 3 OF 32 EUROPATFULL COPYRIGHT 2000 WILA

ABEN An integrated circuit terminal of a data processing system (10) is used to communicate multiplexed signals with an external device. During a reset operation in which a reset signal is asserted, a desired internal clock signal is driven to the integrated circuit terminal such that an emulation system (52) may use the internal clock signal to synchronize an emulation operation. After the reset signal is negated, the emulation system synthesizes the internal clock signal for use during emulation. External visibility of a write operation to a **register** which controls pertinent signal parameters is provided via other integrated circuit terminals when the data processor operates in an emulation mode. The external visibility allows the development system to make similar changes to corresponding signal parameters therein. Therefore, the development system is able to accurately synchronize an emulation operation even when signal parameters are modified during operation. <image>

=> d ab 4

L6 ANSWER 4 OF 32 EUROPATFULL COPYRIGHT 2000 WILA

ABEN An emulation device (11) distributes common control information (8801) to each of a plurality of clock domains (1213,1215, 1217) into which the emulation device is partitioned, and also provides the clock domains with individualized clock control (8905, 8907, 8913). <image>

```
=> d kwic 4


L6    ANSWER 4 OF 32  EUROPATFULL  COPYRIGHT  2000 WILA
DETDEN In addition to testing for functionality and for  manufacturing
defects,
     application software development  requires a similar
       level of simulation, observability and  controllability in the system
or
       sub-system design phase.  The emulation phase of design should
       ensure that an IC  (integrated circuit), or set of ICs, functions
       correctly  in the end equipment.  .  .
       The .  .  .  simulation and continually  increasing cost of CAD
       (computer aided design) tools. In  the board design the side effects
       include decreased  register visibility and control,
       complicated debug and  simulation in design verification, loss of
       conventional  emulation due to loss of physical access by
       packaging  many circuits in one package, increased routing  complexity
       on the board, increased costs of design tools,    mixed-mode packaging,
       and design for produceability. In  application
     development, some side effects are decreased  visibility of
       states, high speed emulation difficulties,  scaled time
       simulation, increased debugging complexity,  and increased costs of
     emulators. Production side  effects involve decreased visibility
       and control,  complications in test vectors and models, increased test
       complexity, mixed-mode packaging, continually.  .  .
       Among the objects of the present invention are to  provide improved
     emulation, simulation and testability  architectures and methods
       which provide visibility and  control without physical probing or
       special test  fixtures; to provide improved emulation,
       simulation and  testability architectures and methods which are
       applicable to critical components of system designs to  support test
and
       integration of both hardware and  software; to provide improved
     emulation, simulation and  testability architectures and methods
       that are a viable  alternative to high capital-cost test equipment and
       systems; to provide improved emulation, simulation and
       testability architectures and methods which integrate  access to
       sophisticated operations in hardware emulation,  fault
     emulation, simulation and built-in test; to provide  improved
     emulation, simulation and testability  architectures and methods
       which apply hardware and  software visibility and control to reduce
     application   development time and thus reduce the
       user manufacturer's time-to-market  on new products; and to provide
       improved emulation,  simulation and testability architectures
       and methods to leverage  hierarchical partitioning and automatically
       generate reusable tests  for related chips and systems..  .  .
       The .  .  .  the adapter  of Figs. 49, 51, 52, 53, 56 and 57;
            FIG. 59A is a compact diagram of shift register  latches
       SRLs in a scan chain in Fig. 59;
            FIG. 60 is a schematic diagram of a code state.  .  .  of clock
       waveforms  for operating the GSP chip of Fig. 80;
            FIG. 83 is a schematic of a parallel register latch for
       use in  the GSP chip of Fig. 80;
            FIG. 84 is a schematic of a serial register latch for use
       in  the GSP chip of Fig. 80;
            FIG. 85 is a block diagram of a control.  .  .
       Extended development system 1101 provides  full-speed, in-circuit
     emulation for system design and  for hardware and software debug
       on widely available  personal computer systems. The development tools
       provide  technological.  .  .  to prototype.  The development system
       elements provide ease of use and  offer the designer the tools needed
to
```

significantly reduce **application** system **development** time and cost to put designs into production faster.

Device . . . configuration file, a GSP (graphic signal processor) configuration, a programmable array logic (PAL) file, an ASIC file and a GPL **register** file.

The . . . software and hardware with the target system. An important emulation interface provides control and access to every memory location and **register** of the target chip and extend the device architecture as an attached processor.

The . . . before the target system is complete. To accomplish this, code is downloaded into the memory on the board 1043 and **executed** at full speed via the interface on an **application** board used in place of the incomplete target system. A suitable **application** board includes a DSP 11, 16K x 32 bits of full-speed (zero wait states) SRAM on a primary bus, two. . .

The . . . Fig. 47 and disclosed in coassigned application SN 057,078 (TI-12033) filed June 2, 1987 and incorporated herein by reference. Shift **register** latches (SRLs) designated "S" are distributed through the device 11 like a string of beads on a serial scan path. . .

Figure . . . TDO communicate with the system. TMS and TCK communicate with a tap controller 1151 which is connected to an instruction **register** 1153 and an instruction decoding circuit 1155.

Test access port (TAP) controller 1151 is in turn coupled to instruction **register** (IR) 1153 and a first multiplexer 1173. The instruction **register** can receive serial scan signals from the TDI line and output serially to MUX 1173. MUX 1173 is under control of the TAP and can select the output signal from the instruction **register** or from another MUX 1171.

The instruction **register** also controls a bypass **register** (BR) 1167 and one or more boundary scan registers (BSR) 1161. The bypass **register** receives the TDI signal and outputs it to MUX 1171. MUX 1171 is under control of the instruction **register** 1153. Based on the instruction loaded into the instruction **register**, MUX 1171 outputs its input from the bypass **register** or its input from one or more BSRs, or internal device **register** scan. Each boundary scan **register** is controlled via the test access port and the instruction **register**.

The . . . or a test mode. During the normal mode, input data entering terminals of IC logic passes through the boundary scan **register**, into the IC logic and out to the normal output terminals without any change due to the BSR. During the. . . test mode, normal input data is interrupted, and test input data is captured, shifted, and updated within the boundary scan **register**. The boundary scan **register** includes two memories, a first memory for receiving and shifting data from the TDI line and a second memory for. . .

Generally, . . . controller card 1141 through pin TDI and enters any one of a number of shift registers, including a boundary scan **register** 1161, a device identification **register** 1163 and design specific test data registers 1165. A bypass **register** 1167 is also provided. These shift registers or serial scan registers are selected via a MUX 1171 under the control. . . output from MUX 1171 is fed to a MUX 1173 so that under control of tap controller 1151 the instruction **register** 1153 or MUX 1171 is selected by MUX 1173. JTAG clock TCK and MUX 1173 output are fed to flip. . .

Returning . . . has two scan paths. One of the scan paths is termed the MPSD data path which usually has numerous shift **register** latches S SRL) serially interconnected like a string of beads throughout module. The second scan path is termed the MPSD control path which generally has fewer shift **register** latches and which selects which MPSD data paths are to be scanned. These scan paths are described in above-cited U.S.. . .

The . . . core domain 1213, system domain 1215 and analysis domain 1217 are shown in Figure 50 and interface through the shift **register** latches of Fig. 47 to all of the various parts of the chip.

Figure . . . chip of device 11. JTAG control 1201 interfaces with the pins via a serial boundary scan assembly including boundary scan **register** 1161 which allows all logic states at the actual pins of device 11 to be read or written. JTAG TAP controller 1151 and JTAG instruction **register** IR 1153 are provided on-chip. Test control 1205 and MPSD control 1203 are integrated into the circuitry. MPSD control 1203. . .

Referring . . . as the following five distinct clocking domains in order to control domain data transfers with the scan clock (JCLK) and **application execution** with the functional clock (FCLK).

1) CPU core domain 1213
2) Analysis domain 1217
3) Peripherals, memory,. . .

There . . . data is scanned to and from the device 11 through internal scan paths that are selected through a JTAG instruction **register** 1153. A unique JTAG opcode for each path allows entry of and access to internal scan data.

Figure 54 shows a further diagrammatic respective of the registers of Figure 50 wherein JTAG instruction **register** IR 1153 is selected for scan between the terminals TDI and TDO. The IR 1153 is decoded in Fig. 50. . . requested by the control card 1141 of Figure 45 via the serial line 1103. These shift registers are the bypass **register** 1167, the boundary scan **register** 1161, the message peripheral 1216 of Fig. 52, the emulation control **register** 1251 and a pair of MPSD scan paths 1252 in the various domains and modules in the domains.

C1 . . . when not scanning and a zero when scanning. C0 and CX are sourced from the emulation control block adapter 1203 **register** 1251. When host computer 1101 detects Ready to Scan for all unlocked domains for a designated device on the target. . .

Connected . . . mode conditioned stop logic circuitry 1309S, 1309C and 1309A in the domains respectively. The modes are established by a mode **register** 1311 which is scanable in Figs. 54 and 57 to establish the type of stop and any other desired mode. . .

Then . . . sent to the CPU domain 1213 to make it stop according to the stop mode established for it in mode **register** 1311 and mode conditioned stop circuitry 1309C. DONE circuitry 1363 detects when the stop is completed and signals back to. . .

Fig. . . . an example of process steps by which the scan control 1149 including IR 1153, adapter 1203 including ECR (emulation control **register**) 1251, and host computer 1101 cooperate to enter and perform sequences of commands on-chip.

Next . . . to reach the state "Select-DR-Scan". This means that the scan controller is ready to accept scan into a DR (data **register**) identified just before as the ECR 1251 by the host to the IR 1153. Into ECR 1251 the host 1101. . .

In Figure 59, serial scan bits enter the emulation control **register** ECR 1251 which is subdivided into a shift **register** LOCK 1351 for holding bits to lock and unlock domains, a first C0,CX control code shift **register** named CODA 1353, a second C0,CX control code shift **register** named CODB 1357, a shift **register** 1359 associated with event manager circuitry 1365, and a two-bit **register** JMODE 1360. These registers are compactly il trated in Fig. 59A. Thus, the .al scan enters on a scan line SIN, . .

A code state machine 1381 controls a two input MUX 1383. MUX 1383 selects the C0,CX two-bit contents of shift **register** 1353 or 1357 and loads them into an enabled one of three flip flops 1393, 1395 and 1397. A lock control circuit 1401 operating under the control of lock shift **register** 1351 and code state machine 1381 sends lock signals to disable or enable each of the flip flops 1393, 1395.

.

.

Test . . . the test codes on line 1421, thus overriding the code state machine feature. This option is selected by scanning JMODE **register** with "00" (both bits zero). Thus, the preferred embodiment is accommodates direct host control of the domains, wherein the latter. . .

In . . . and CODB 1357 as directed by a code state machine 1381;

2) Program state machine 1381 operations via REVT (**Register** Event) **register** 1359 to respond to stimuli including:

a) START from scan control 1149

b) DONE from CPU core 1213,. . . to JCLK (with a code 00 in both registers 1353 and 1357) and vice versa for each domain, via LOCK **register** 1351; and

4) Lock domains in their current state while operating with other domains.

The operation of code state machine 1381 is now further described. When the JTAG IR (Instruction **Register** 1153) is loaded with a scan path select command for path 1251, a line ECRSEL feeds a signal to state. . . whereupon the state machine 1381 enters a lock state. This allows the registers 1351, 1353 and 1357, the event manager **register** 1359, and JMODE **register** 1360 to be changed without disturbing the MPSD codes and clocks supplied from flip flops 1393, 1395, 1397 and selection circuits 1371, 1373 and 1375 to the domains 1215, 1213 and 1217. Lock **register** LOCK 1351 holds bits which selectively cause the CPU, analysis and system domains 1213, 1217 and 1215 to be frozen. . .

A NAND gate 1471 supplies a SET input of SRL 1451 in response to two inputs ECRSEL (emulation control **register** select) and START from Fig. 50. An AND gate 1473 supplies a RESET input of SRL 1451 in response to. . .

Shift **register** 1359 includes scannable **register** -event bits REVT3, REVT2, REVT1 and REVT0. REVT3 is a scanable bit inverted by an inverter 1483 to signal a NAND. . . to CNTBRW, DONE and EMU1. Additional NAND gates for EMU0 and other signals are readily providable, as are further event **register** cells in shift **register** 1359.

The . . . MPSD control code currently applied to the domains when the emulation control path ECR 1251 is selected by the instruction **register** IR 1153 and a START pulse is generated. The code state machine 1381 exits the lock state upon a START. . .

The . . . registers 1353 and 1357 selected by the code state machine 1381. When the code state machine 1381 points to a **register** CODA or CODB containing a different clock source for that state, the clock switch 1411, 1413 or 1415 corresponding to the unlocked domain selected by LOCK **register** 1351 and lock control 1401 locks the code of the old state by signaling NOT GCTD (Not Good Clock This. . .

In Fig. 62, selecting circuit 1383 has two MUXes 1501 and 1503 for respectively selecting the C0,CX control code from **register** CODA or CODB in response to signal LSB from the code state machine 1381 of Fig. 59. The C0,CX codes in **register** CODA are designated

CODA0 and CODAX, and in **register** CODB are CODB0 and CODBX. The selected code is held in a pair of SRLs 1507 for C0 and 1509. . .

In . . . OR gates 1531, 1533 and 1535 each having first inputs connected respectively to lines LOCKRQS, LOCKRQC and LOCKRQA from LOCK **register** 1351 of Fig. 59, and outputs connected respectively to lines LOCKS, LOCKA and LOCKC of Figs. 59 and 62. These. . . 1455 of Fig. 60. The AND gate 1543 has two inputs connected to SWINPROG and nclksel. In this way, LOCK **register** 1351 bits override any other signals when the bits call for locking the domain (test clock JCLK only). However, if **register** 1351 unlocks any one or more domains (calling for functional clock FCLK to each such domain), each such domain can. . .

In . . . and 1565. This block 1205 is enabled by a low active OR 1571 supplied from the two bits of JMODE **register** 1360 of Fig. 59. Code conversion logic block 1205 thus converts to MPSD code from any three-wire testability code scheme. . .

In . . . the operational mode of the device for the entire session. The operating mode is specified in a separate emulation mode **register** or module 1311 which is scan-loaded while the CPU core is halted. This advantageous mode feature involves mode driven stops.. . .

The particular stop mode of the processor is determined by mode **register** 1311 of Fig. 57, as already discussed. The location, placement in any Particular **register**, and scanability of each of the bits of the stop mode is quite flexible. In another example, the particular stop. . . If the stop mode is generally not changed frequently in practice, then it can be put in a separate mode **register** 1311 to avoid unnecessary scan bits in the various domains. Otherwise, the various stop mode bits can be allocated to. . .

In . . . transferred to initiate a memory or I/O transaction, especially when the memory transfer is a single word. When the entire **register** file may be used to create a block transfer, multiple scan modules may be used.

Advantageously, . . . confers a level of concurrency beyond more JTAG boundary scan. Microprocessors, for one example, are a very valuable and complex **application** with access to the internal information very important. Since JTAG boundary scan involves a test port, this test port is. . . communication to specific chips one at a time or all together, concurrently. The preferred embodiment puts commands in emulation control **register** 1251 and loosely couples the communication so that the device 11 can run in real time when desired. In this. . . to the commands such as CODA and CODB in Fig. 59, in contrast to loading the JTAG IR 1153 and **executing** an operation by decode.

Thus, the bits which are scanned in are loaded into a 12 bit **register** 1707, 3 bit **register** 1709 and single bit registers 1711.1 through 1711.8 in this section of the analysis domain 1217.

The three bits in shift **register** 1709 make a one of eight selection by MUX 1713 and supply the selected line to a 12 bit down. . . counter 1715. A predetermined count is loaded into the J (jam) parallel input of counter 1715 from 12 bit shift **register** 1707. As signals on the line selected by MUX 1713 occur, the 12 bit down counter counts down until the number represented by the contents of shift **register** 1707 is exhausted, whereupon a borrow line 1717 goes active and is fed to selector 1703.1. The borrow line signal. . .

An . . . utility of the down counter 1715 (besides single-stepping) is as follows. Assume that the counter 1715 is set by scan

register 1707 200 and the MUX 1713 is set b **register** 1709 to select BPDA data address breakpoint. The particular data address is scan-entered in a **register** 1813 in Fig. 71. This configuration stops the processor after the specific scan-identified data address has been addressed 200 times.. . .

In another filter example, the counter borrow line is selected by scan **register** 1711.1 and fed out of pin EMU0 to permit external logic to count events at a rate stepped down by. . .

When external logic is used, its resolution is equal to the number set for the counter 1715 by **register** 1707. Full resolution is obtained by reading out the value in the 12 bit down counter, allowing determination of the. . .

In . . . step 1737, a trace stack full condition is sensed. Then a step 1739 selects which conditions are relevant using shift **register** 1711 and logic 1703 of Fig. 69 for example . Of the conditions selected, a count is kept in step. . .

In . . . bus 101A of Fig. 68A is connected to a digital comparator 1811. A reference value is scan-loaded into a further **register** 1813 in the analysis domain having most significant bits MSB and least significant bits LSB. When a program address asserted on address bus 101A is identical to the contents of **register** 1813, then comparator 1811 produces an output indicative of a breakpoint address occurrence on line BPPA.

In . . . any address within a selected group of addresses such as the ranp;, indicated by the most significant bits MSB of **register** 1813. In such case, a scanable mask **register** LSBEN is scan-loaded to disable the response of comparator 1811 to the LSB bits of **register** 1813. Only the most significant bits are compared by comparator 1811 in this mask condition, thereby providing a breakpoint on. . .

Scanable **register** 1813 for breakpoint purposes requires no connection to data bus 111D. However, this **register** 1813 is advantageously reused for message passing access between the emulation/simulation/test host computer 1101 of Fig 45 and the data. . . of the target device. The message passing function is used when breakpoint sensing need not occur, and vice-versa, so that **register** 1813 feasibly performs different functions at different times.

In . . . address discontinuities in operation of program counter 93 of Fig. 68A in the CPU core domain. A scanable trace stack **register** section 1823 responds to control circuit 221 to push a program counter 93 address value onto the trace stack when. . .

The state vector leading bits of the program counter trace stack 1821 resemble a shift **register** for stack purposes and the S/E9 end of the shift **register** is output to an OR gate 1825. OR gate 1825 is further connected to a program count shift out **register** PCSO 1827. The output of PCSO 1827 is fed back to a second input of OR gate 1825. When a first logic one is pushed from the bottom of the stack 1821 into OR gate 1825, **register** 1827 indicates the output trace stack full TRFUL for analysis circuitry of Fig. 69. Subsequent push onto stack 1823 pushes out a subsequent zero from the stack into OR gate 1825. However, **register** 1827 continues to be loaded by OR-gate 1825 (by virtue of the feedback from PCSO to 1825) with a one. . .

Figure . . . in the device 11. A serial scan path of Figs. 54 and 78 has serial data MSIN enter a shift **register** 1923 CMD/STATUS for entry of commands to operate the message passing circuitry 1216 and for scan out of status information. The serial path continues to a further seri **register** 1925 designated 16 B DATA **REGISTER** whereupon the scan path exits on a line designated MSOUT. These shift registers correspond to serial/parallel interface 1881 of Figure 76. The function of shift **register** 1925 can be implemented by **register** 1813 in the analysis domain and reuse principles can generally minimize the chip real estate required for message passing.

The shift **register** 1925 is connected to the output of a MUX 1931 which selects one of three paths to load into the **register** 1925. Two of these paths are the data and address portions of the TIBUS peripheral bus of Fig. 52 and. . . portion is designated 1935 and the address portion is designated 1937 in Fig. 78. The third path called the communication **register** bus 1939 is connected to a communication **register** 1941.

The . . . circuitry 1216 is further described in connection with an example of simulated peripheral accesses. The device 11 suitably parallel-loads the **register** 1925 via MUX 1931 when a peripheral access or other outbound communication is commenced. Host computer 1101 scan up-loads the contents of **register** 1925, and then determines the expected response of the absent peripheral by simulation computations. Host computer 1101 then scan down-loads the simulated response of the absent peripheral into the **register** 1925. This information in **register** 1925 includes the data which would be returned from the absent peripheral in response to a Read. To convey the data to the peripheral bus, **register** 1925 is selected by a MUX 1945 to be loaded into a communication **register** 1941. Communication **register** 1941 then supplies the data through a MUX 1955 and then an output buffer 1947 onto the data bus portion. . .

MUX . . . communications outward bound on TI data bus 1935 reach MUX 1945 at an input 1951 and are communicated via communication **register** 1941 through communication **register** bus 1939 and MUX 1931 to the 16-bit data **register** 1925.

MUX 1955 selects either the communication **register** bus 1939 or an additional bus 1961 directly connected to data **register** 1925. In this way, data can be even more directly communicated from **register** 1925 via path 1961, MUX 1955 and output buffer 1947 to the TI data bus.

Buffer status flags are communicated from hardware 1965 of device 11 along with Read/Write- signal R/W- to CMD/STATUS **register** 1923 for scan out to host computer 1101. The host computer receives these buffer status flags and returns reply command. . .

Some of the command bits from **register** 1923 are communicated to a command decoder CMD DEC 1971. Decoder 1971 decodes the commands and selectively activates operation output. . . aspects, MUX 1945 has an input 1951 connected to the data portion of the peripheral bus TIBUS for further flexibility. **Register** 1925 is connected to interrupt generation block 1943 so that even the interrupt status of device 11 can be scan. . .

It is to be emphasized that functional clock FCLK operates when data is loaded into **register** 1925 from the device 11 peripheral bus and when buffer status flags are loaded into **register** 1923.

Test clock JCLK operates when the data in registers 1923 and 1925 are scan up-loaded to host computer 1101, and when data is scan down-loaded to these two registers. Then functional FCLK operates to send data from **register** 1923 to command decoder 1971 and to send data from **register** 1925 to the MUXes, registers and buffers and buses of the message passing circuitry and the rest of device 11.. . .

A **register** UID 1981 is connected to the data bus 1935. A further **register** JID 1983 is connected to the data **register** 1925. The outputs of registers UID and JID are supplied to ask identification compare ci it 1985. When the identificati . . .

In . . . occurs, the monitor communicates with the emulation host computer 1101 through a TIBUS peripheral such as message passing circuitry 1216 **register** 1925 having an address that resides in the TIBUS address space. Once a trap has been taken and until a. . .

A . . . the address and the type of access (read or write). The emulation controller 1101 then provides the data through a **register** 1925 on reads or reads the data directly off the data bus on writes. READY signals for the completion of the cycle are also provided serially through the scan path to **register** 1923. After the appropriate transfers take place, the CPU core of device 11 is restarted.

The . . . provided task ID is implemented through the Message Passing Peripheral. The user's program provides a task ID through a TIBUS **register** value. This value is compared to a value loaded via scan. The comparison is enabled via an extra bit which. . .

The **register** 1923 in Fig. 78 is implemented as a 3 bit opcode, a four bit status field, and a nine bit. . .

The . . . 1943 of Fig. 78.

3. The TIBUS peripheral bus can be programmed by scan to Read and Write to **register** 1925 when no frame recognizes a select.

4. The TIBUS peripheral bus can be programmed via scan to stop. . . the core and device with the second cycle of the bus active, allowing the host 1101 to load or unload **register** 1925 and obtain address and a read/write indicator.

5. Restart the device execution from the message passing circuitry 1216 **register** 1923.

Each . . . the TI bus address and the TI bus block is addressed during simulation pipe freeze with simulated peripheral access in **register** 1923 enabled, the core stops before the second cycle of the peripheral access is complete. In this mode, reads are. . . no other frame is decoded. All writes are directed at the message peripheral. When the simulated peripheral access bit in **register** 1923 is enabled, four status bits are used to specify the number of wait states associated with the peripheral access.. .

In category 1 the embedded microcontroller CPU **execution** ceases for **application** purposes and the CPU transfers data, for example, to the host computer 1101. Host computer 1101 does memory reads and. . .

In Category 2 (concurrent **execution**) a message passing peripheral MSGPASS 1216 of Figs. 52, 54 and 78 is included in the preferred embodiment combination. Advantageously MSGPASS 1216 allows the microcontroller to **execute** other tasks after calling the host computer 1101 for service. Then code from the **application** system is sent via the scan serial line 1103, and inserts interrupts over EMU0 line to software control to make. . .

Block transfers are accomplished by use of the message passing peripheral 1216 by loading the **register** 1925 from communication **register** 1941, and with JTAG controller in the IDLE state, do N (e.g 16) bit serial shift, and then cycling back to load **register** 1925 again. The block transfers can be directed to any other serial interface to which the scan path is connected.. . .

The . . . is advantageously further useful for development system purposes. Host computer 1101 operating as a development system downloads a command to **register** 1923 requesting the machine state of the embedded microcontroller. The microcontroller responds by trapping (analogous to an interrupt) to prestored. . .

A . . . of a GSP chip 2120 having a central processing unit 2200 connected by buses 2202, 2204, 2206 and 2208 to **register** files 2220, instruction cache 2230, host interface 2240 and graphics hardware 2210 respectively. A further bus 2205 interconnects a host.

.

For . . . testability, the GSP 2120 memory elements are split into two types:

(1) Multiple-bit registers such as those in the **register** file 2220, the Cache RAM 2230, a memory address **register** 2103, memory data **register** 2105 and a field size **register** 2107. These are all on wide buses, and sufficient logic is included to ensure that there is a route from. . . these registers to local address data (LAD) pins of the chip.

(2) Serial latches, such as an emulation control **register** 2121, buffer SRLs 2135 of a control ROM (CROM) 2131, and scanable registers of core processing circuitry 2101 on chip.. . .

One approach to testability herein is called parallel serial scan design (PSSD). A rule is imposed in which every **register** bit and serial latch are only loaded by some function ANDED with, or conditional on, a single clock phase (H3T). . .

Each parallel **register** cell has a circuit shown in Fig. 83. It is loaded on the (normally conditional) H3T phase and is sampled on a (conditional) H1T phase. Control logic and microcode are included on-chip to enable every parallel **register** to be loaded and dumped onto the LAD bus. Thus, the machine state can be loaded up, executed, and then. . .

A circuit for each serial latch is shown in Figure 84. It is similar in form to the parallel **register** circuit of Fig. 83, but contains an additional serial input called the scan input Scan In. For the purposes of. . .

A memory address **register** MA 2103, a memory data **register** MD 2105 and a field size **register** 2107 are associated with main core 2101. A four wire scan interface or port 2111 is connected to a selecting. . . pins. Emulation control pins EC0 and EC1 provide further control inputs. Selecting circuit 2112 is connected to an emulation control **register** 2121. **Register** 2121 is also called a scan control **register** herein. Special test TST and Compress COM bits 0 and 1 in **register** 2121 are connected to selector circuit 2112 to route the lines 2115 and 2117 to one of three scan paths. The first path allows scan to **register** 2121 itself. The second path allows scan of CROM buffers 2135 and core 2101. The third path connects to a.

. .

A **register** select code is supplied by emulation control **register** 2121 bits 2-5 SCAN SEL on a line 2125 to control the MUX 2113. In this way, **register** selection of a selected one of registers 2103, 2105 and 2107 is controlled by emulation control **register** 2121 in its operation of MUX 2113. Scan data in and data out on lines 2115 and 2117 are thus. . .

Test modes are controlled via the EC1, EC0 and SCIN pins, and two bits TEST and COMPRESS of the emulation control **register** 2121. The . . . define the state of the emulation control port. Scanning a 1 into the TEST bit zero (0) of the control **register** 2121 redefines the port as a test control port as long as a code 111 (for EC1, EC0 and SCIN). . . to the interface 2111. The 111 code is a normal user run mode and also clears the entire emulation control **register** 2121, including the TEST bit, thus resetting the port. The relationship of the codes to the MPSD codes tabulated earlier. .

.

In Fig. 81 control ROM (CROM) 2131 is connected to main CPU 2101. The second bit COMPRESS of the emulation control **register** 2121 extends the possible number of test states available via the interface and is used for CROM compressions in a type testing called signature analysis. In such signature analysis, a scanable linear feedback shift **register** 2141 of Fig. 86 is combined with the CROM buffers 2135 of Figs. 81 and 85 and is used to. . .

In Fig. 86, the scanable linear feedback shift **register** 2141 utilizes a data compression method of self-testing the CROM. The method accesses every state. For each access, a word. . .

A stack **register** 2145 in normal operation is used during micro-state pops and pushes in the circuit of Fig. 85. In CROM test mode this **register** 2145 is reused as a 13-bit counter. The two most significant bits of the counter are used to control whether. . .

The . . . compression on CROM output:

SCANCOMP - PLOAD - SCAN(I) - CROM - SCAN(O)

SCANTEST means "Scan TEST bit into control **register**".

SCANCOMP means "Scan TEST and COMPRESS bits into control **register**".

SCAN is a simultaneous scan in and scan out. Scan(I) implies the scan is scan-in for data initialization. SCAN(O) implies. . .

In . . . 111 is applied for two cycles to exit the test mode and clear the entire contents of the scan control **register**. If the TEST bit is set, the first cycle clears only the test bit. In the second cycle, the code 111 and the cleared TEST bit then clears the rest of the scan control **register** 2121. In this way, exit occurs from the test mode into an emulation mode such as an emulation controlled run.. . .

Using . . . modification, inspect and/or alter, while the processor is in stop mode without change to user environment. This includes internal (I/O **register**) and external memory spaces.

Inspection and/or modification while in stop mode is also performed. Modification of all internal registers including PC (program counter), ST(status **register**), and SP (stack pointer) while in stop mode is also available.

3. Single stepping of instructions.

Internal state information utilized by the emulator host computer 1101 of Fig. 45 includes the program counter PC, the **register** file of CPU 2101 and cache, segment registers and p-flags.

The . . . at the pins EC1, EC0 and SCIN are as follows: Normal functional mode, controlled execution mode, halt, pause, emulation control **register** scan and scan of data registers 2103, 2105 and 2107. See Table VI. These codes are essentially the MPSD codes. . .

All . . . registers are scanned in through the LSB (least significant bit) and out through the MSB (most significant bit). Emulator control **register** 2121 is scannable in response to the Table VI code "emulation control **register** scan" applied to pins EC1 and EC0.

The . . . power up. Reset is a global signal within the chip. The functional run mode code continuously clears the emulation control **register** 2121.

The . . . such as load and dump. In this mode, reset is gated with a Block Reset bit in the emulation control **register**. The emulation control **register** 2121 is not cleared on reset in this mode.

The . . . initiated memory cycles to complete. Second, the CPU stores the contents of the program counter PC in the memory data **register** 2105. Third, the CPU signals the stop by generating a STOPACK signal, forcing scan out line SCOUT low. Fourth, the. . .

The scannable registers are memory address **register** 2013, memory data **register** 2105 and field size **register** 2107. **Register** 2103 and 2105 each have 32 bits, For example, the field size **register** 2107 is 6 bits. The **register** to be scanned is determined by a data scan select field (bits 2-5) of the emulation control **register** 2121.

The microcode in the CROM accomplishes four main functions on command. First, it transfers data from a selected device **register** or from cache or from program counter to the memory data MD **register** 2105. Second, it transfers from the MD **register** 2105 to a selected device **register** or to cache or to program counter. Third, it executes MPSD code to do step-by-step operations. Fourth, it sends instructions to the memory interface 2250 to transfer data either way between external memory and

the **register** pair MD 2105 and MA 2013.

Memory address **register** 2103 holds the address for all CPU initiated memory accesses including those of the emulator. After a memory access is completed, **register** 2103 is incremented by 32 bits to point to the next word address. The low 5 bits are left unchanged.. . . contents are loaded into the upper 28 bits and the halt condition code occupies the low four bits of this **register**

.

Memory data **register** 2105 passes data between the emulator and memory controller. **Register** 2105 serves as a data latch for passing data between the emulator and the CPU for loads and dumps. When the CPU is initially halted, this **register** 2105 contains an image of the CPU program counter and a halt code in the low order four bits. This. . .

The field size **register** 2107 makes it possible to do memory accesses to data fields of various sizes specifiable by the field size **register**.

The value loaded into **register** 2107 is a 6 bit code that indicates the number of bits to be written. Using **register** 2107 permits the emulator to write to bits or fields without having to do a read-modify-write operational sequence, which could. . . host computer access operations occurring between the read and write of the sequence. When the CPU is initially halted, the **register** 2107 contains an indeterminate value, since it is the actual latch and not a

copy. Scan in of a value into **register** 2107 enters the value in the most significant 6 bits of **register** 2107. Upon scanning out the value, the value is in the least significant 6 bits of **register** 2107.

The emulation control **register** has bits as specified in Table IX. The contents of emulation control **register** 2121 are not executed until control scan mode is changed to another MUX. <image> <image> <image>

Four bits 2-5 in the emulation control **register** 2121 select one of the registers 2103, 2105 and 2107 for serial scan during Data **Register** Scan mode. During normal functional mode, these four bits are cleared to zero. Table X shows the scannable registers and.

.

.

The . . . a set of functions that are tabulated in Table XI. The proper function code is placed in the emulation control **register**. The processor CPU 2101 is then placed in the controlled run mode. The CPU 2101 then forces line SCOUT high,. . .

The bits FCN3, FCN2, FCN1 and FCN0 occupy bits 9-6 of the emulation control **register** 2121.

In . . . dump. The CPU 2101 then fetches the requested parts of the machine state and loads them into the memory data **register** 2105 one by one. For each group of 32 bits, the emulator host computer 1101 scans out memory data **register** 2105 serially to obtain the data. More specifically, the operation for emulator dump is as follows. First, the emulator scans. . . Table XI to dump the state using the control scan mode and sets the emulator busy enable bit 10 of **register** [illegible]1. Second, the emulator enters e controlled execution mode. Third, the CPU 2101 forces SCPOUT pin high. Fourth, the CPU 2101 places a 32 bit word of the machine state in **register** 2105 and forces SCOUT pin low. Fifth, CPU waits for the cycle to complete. When emulator busy bit is enabled,. . . signals stop acknowledge STOPACK on the SCOUT pin low. Sixth, the emulator enters a data scan mode and scans the **register** 2105. When scanning, the CPU 2101 is inhibited from concurrent activity in this embodiment. Seventh, operations return to step two. . .

The end of the process is determined by the known number of words to dump. After dumping the cache and the **register** file, an extra controlled run is executed after the last word has been scanned out so that the CPU can. . .

The emulator then clears the emulator busy bit in the emulation control **register** 2121.

For example, in the function DUMP ST,PC, the "1000" function code causes the CPU program counter and status **register** to be dumped. The status **register** is dumped first, followed by the PC.

In the DUMP REG.FILE function, the "1001" function code causes the A and B **register** files to be dumped in that order.

In . . . load functions, designated by codes 1011, 1100 and 1101, the emulator scans in a load request and values into the **register** 2105. CPU 2105 then builds the machine state from values in **register** 2105. First, the emulator scans in the code to load the state using the emulation control **register** scan mode and then sets the emulator busy bit. Second, the emulator scans in **register** 2105 using the data **register** scan mode. Third, the emulator enters the controlled execution mode, and fourth, the CPU 2101 forces line SCOUT high. Fifth,. . . the data has been loaded, line SCOUT is forced low. Sith, the emulator scans a succeeding 32 bit word into **register** 2105. Exiting the data scan **register** mode clears the busy flag. Seventh, operations return to the third step of entering the third execution mode. The end.

. . load which is a predetermined number. The emulator then clears the emulation busy enable bit 10 in the emulation control **register** 2121.

In the LOAD PC,ST function, the 1011 function code causes the status **register** to be loaded followed by the CPU program counter. In the LOAD REGS 1100 function code, the A and B **register** files are loaded in that order. In the LOAD CACHE function 1101 code, the cache is loaded starting with segment. . .

The emulator can access any part of the chip address space including I/O registers by scanning in address values to **register** 2103 and data values to **register** 2105, together with a memory read or write function code to emulation control **register** 2121.

When the CPU is in the emulator halt state, these registers are available to the emulator and the controlled. . . code using this mechanism, the emulator flushes the cache by setting the cache flush bit 29 in the emulation control **register** 2121.

The functions of the bits of emulation control **register** 2121 (which is analogous to emulation control **register** 1251 of Fig. 59) are now discussed in even further specific detail. If emulation busy enable bit 10 is set. . .

Load . . . During normal functional mode, this bit is cleared to zero. The emulator insures that the "data" part of the MA **register** 2103 contents is not contained in the least significant five bits as these bits of the **register** 2103 are not output to the LAD bus. The emulator insures that the least significant five bits are loaded with. . .

A single step control bit 16 in the emulation control **register** 2121 causes core 2101 to execute only one instruction before generating

```
          a stop acknowledge STOPACK  signal on the SCO    pin.  .  .
instruction
          into the instruction stream after the current  instruction. This bit
          ORed with the single step bit in  the status register before
          going to the microcontroller.  During normal functional mode, this bit
          is cleared to  zero.
          A .  .  .  CPU 2101 inserts a code 1111 (halt due to  bus fault) on the
          least significant four bits of memory  address register 2103,
          along with the contents of the  program counter.
          A single step mode bit 16 in the emulation control  register
          and a similar bit in CPU 2101 status register,  control CPU
          function for single step. When either of the  single step bits are set
          to "1", the CPU halts.  .  .
          Interrupt logic associated with core 2101 monitors  for interrupts
          regardless of the state of the emulation  control register
          2121. Thus, the state of an INTPEND IO  register will be the
          same as if the core 2101 has  interrupts masked for any HALT/SCAN
          periods. On a  transition into run or single step with the interrupt
          enable bit of the status register set, the highest  priority
          pending interrupt is taken. In this way, the  interrupt enable bit is
          cleared of status which.  .  .
          The .  .  .  1203. In addition, as will be  apparent from the following
          description, the lock control circuitry  at 1401 and the LOCK
        register 1351 can be eliminated from the  adaptor 1203. As will
          also be apparent from the following  description, any two of.  .  .


=> d ab 5


L6      ANSWER 5 OF 32  EUROPATFULL  COPYRIGHT  2000 WILA


=> d ab 5


L6      ANSWER 5 OF 32  EUROPATFULL  COPYRIGHT  2000 WILA


=> d ab 6


L6      ANSWER 6 OF 32  EUROPATFULL  COPYRIGHT  2000 WILA


=> d ab 7


L6      ANSWER 7 OF 32  EUROPATFULL  COPYRIGHT  2000 WILA


=> d ab 8


L6    ANSWER 8 OF 32  PROMT  COPYRIGHT 2000 IAC

AB    At the Windows Hardware Engineering Conference (WinHEC) in San Francisco
      last month, Microsoft demonstrated a virtual racing car. Like a
      ground-borne aircraft flight simulator, this machine would give drivers
      all the visual and auditory excitement of a race car turning the track at
      160 or 180 mph-as well as the corresponding drags, pulls and bumps in the
      steering column and pedals. "This would be useful for race car driver
      training," a smiling Bill Gates told an audience of 4,000. And in the
      WinHEC exhibition, aisle after aisle of hardware manufacturers
      demonstrated what they were doing to promote similar entertainments on
```

```
     Windows platfo     It was a virtual funhouse of    zling 3-D graphics
     effects, streamıng video windows and dramatic positional sound effects.
     The demonstrations at WinHEC seemed to verify that home
     entertainment-Windows games and Digital Versatile Disk (DVD) playback
     systems-would still be the primary application for analog and
mixed-signal
     ICs developed to serve audio, video and multimedia. While 3-D data
     visualization for financial analysis might be appropriate business
     applications of 3-D graphics, suggested graphics industry analyst John
     Peddie Associates (Tiberon, Calif.), it will probably be 2005 before 3-D
     graphics and animation are used outside of PC games and movies.
     Similarly, Mike Feibus of Mercury Research (Phoenix) sees business audio
     applications as a distinct possibility for 1998. But few people now have
a
     clear idea of what the most dramatic business applications would be.
While
     3-D graphics could be used for data visualization for financial analysis,
     suggested Chris Henningsen, a marketing manager with Harris Corp.
     (Melbourne, Fla.), that might depend on multimedia's ability to record
     audio and video, to packetize it for delivery over phone lines, corporate
     networks or the Internet. Thus, there is related development in video
     capture devices, A/D converters for audio, or, in Harris' case,
     videoconferencing components. But the bulk of graphics, video and audio
     controllers are being developed to support a $1.2 billion Windows games
     industry.
     The most dramatic of these mixed-signal devices are "accelerators" for
     both 3-D graphics and audio, which offload multimedia processing tasks
     from the host CPU.
      THIS IS AN EXCERPT: COPYRIGHT 1997 CMP Publications Inc.


=> d ab 9


L6   ANSWER 9 OF 32  USPATFULL
AB     The invention provides improved architectures and methods for
emulation,
       simulation, and testability of data processing devices and systems
       without requiring physical probing or special test fixtures. A data
       processing device may include a semiconductor chip that is divided into
       domains. One domain may be halted and tested while another domain
       continues to operate. For example, the semiconductor chip may have a
       electronic processor domain and an analysis domain. The analysis domain
       may include an on-chip condition sensor that is connected to the
       electronic processor domain. The chip can further include control logic
       circuitry to allow the analysis domain to operate while the electronic
       processor is halted and tested.


=> d kwic 9


L6   ANSWER 9 OF 32  USPATFULL
SUMM   In addition to testing for functionality and for manufacturing defects,
     application software development requires a similar
       level of simulation, observability and controllability in the system or
       sub-system design phase. The emulation phase of design should
       ensure that an IC (integrated circuit), or set of ICs, functions
       correctly in the end equipment.  .  .
SUMM   .  .  .  simulation and continually increasing cost of CAD (computer
       aided design) tools. In the board design the side effects include
       decreased register visibility and control, complicated debug
       and simulation in design verification, loss of conventional
     emulation due to loss of physical access by packaging many
       circuits in one package, increased routing complexity on the board,
```

increased cost of design tools, mixed-mode packaging, and design for produceability. In **application development**, some side effects are decreased visibility of states, high speed **emulation** difficulties, scaled time simulation, increased debugging complexity, and increased costs of **emulators**. Production side effects involve decreased visibility and control, complications in test vectors and models, increased test complexity, mixed-mode packaging, continually. . .

SUMM Among the objects of the present invention are to provide improved **emulation**, simulation and testability architectures and methods which provide visibility and control without physical probing or special test fixtures; to provide improved **emulation**, simulation and testability architectures and methods which are applicable to critical components of system designs to support test and integration of both hardware and software; to provide improved **emulation**, simulation and testability architectures and methods that are a viable alternative to high capital-cost test equipment and systems; to provide improved **emulation**, simulation and testability architectures and methods which integrate access to sophisticated operations in hardware **emulation**, fault **emulation**, simulation and built-in test; to provide improved **emulation**, simulation and testability architectures and methods which apply hardware and software visibility and control to reduce **application development** time and thus reduce the user manufacturer's time-to-market on new products; and to provide improved **emulation**, simulation and testability architectures and methods to leverage hierarchical partitioning and automatically generate reusable tests for related chips and systems.

DRWD FIG. 16A is a compact diagram of shift **register** latches SRLs in a scan chain in FIG. 16;

DRWD FIG. 40 is a schematic of a parallel **register** latch for use in the GSP chip of FIG. 37;

DRWD FIG. 41 is a schematic of a serial **register** latch for use in the GSP chip of FIG. 37;

DETD Extended development system 1101 provides full-speed, in-circuit **emulation** for system design and for hardware and software debug on widely available personal computer systems. The development tools provide technological. . . to prototype. The development system elements provide ease of use and offer the designer the tools needed to significantly reduce **application** system **development** time and cost to put designs into production faster.

DETD . . . configuration file, a GSP (graphic signal processor) configuration, a programmable array logic (PAL) file, an ASIC file and a GPL **register** file.

DETD . . . software and hardware with the target system. An important emulation interface provides control and access to every memory location and **register** of the target chip and extend the device architecture as an attached processor.

DETD . . . before the target system is complete. To accomplish this, code is downloaded into the memory on the board 1043 and **executed** at full speed via the interface on an **application** board used in place of the incomplete target system. A suitable **application** board includes a DSP 11, 16 K.times.32 bits of full-speed (zero wait states) BRAN on a primary bus, two selectable. . .

DETD . . . coassigned application Ser. No. 06/057,078 (TI-12033) issued Aug. 22, 1989, (U.S. Pat. No. 4,860,290) and incorporated herein by reference. Shift **register** latches (SRLS) designated "S" are distributed through the device 11 like a string of beads on a serial scan path. . .

DETD . . . TDO communicate with the system. TMS and TCK communicate with a tap controller 1151 which is connected to an instruction

register 1153 and an instruction decoding circuit 1155.

DETD Test access port (TAP) controller 1151 is in turn coupled to instruction **register** (IR) 1153 and a first multiplexer 1173. The instruction **register** can receive serial scan signals from the TDI line and output serially to MUX 1173. MUX 1173 is under control of the TAP and can select the output signal from the instruction **register** or from another MUX 1171.

DETD The instruction **register** also controls a bypass **register** (BR) 1167 and one or more boundary scan registers (BSR) 1161. The bypass **register** receives the TDI signal and outputs it to MUX 1171. MUX 1171 is under control of the instruction **register** 1153. Based on the instruction loaded into the instruction **register**, MUX 1171 outputs its input from the bypass **register** or its input from one or more BSRs, or internal device **register** scan. Each boundary scan **register** is controlled via the test access port and the instruction **register**.

DETD . . . or a test mods. During the normal mode, input data entering terminals of IC logic passes through the boundary scan **register** , into the IC logic and out to the normal output terminals without any change due to the BSR. During the. . . test mode, normal input data is interrupted, and test input data is captured, shifted, and updated within the boundary scan **register**. The boundary scan **register** includes two memories, a first memory for receiving and shifting data from the TDI line and a second memory for. . .

DETD . . . controller card 1141 through pin TDI and enters any one of a number of shift registers, including a boundary scan **register** 1161, a device identification **register** 1163 and design specific test data registers 1165. A bypass **register** 1167 is also provided. These shift registers or serial scan registers are selected via a MUX 1171 under the control. . . output from MUX 1171 is fed to a MUX 1173 so that under control of tap controller 1151 the instruction **register** 1153 or MUX 1171 is selected by MUX 1173. JTAG clock TCK and MUX 1173 output are fed to flip. . .

DETD . . . has two scan paths. One of the scan paths is termed the MPSD data path which usually has numerous shift **register** latches S (or SRL) serially interconnected like a string of beads throughout the module. The second scan path is termed the MPSD control path which generally has fewer shift **register** latches and which selects which MPSD data paths are to be scanned. These scan paths are described in above-cited application. . .

DETD . . . core domain 1213, system domain 1215 and analysis domain 1217 are shown in FIG. 7 and interface through the shift **register** latches of FIG. 4 to all of the various parts of the chip.

DETD . . . chip of device 11. JTAG control 1201 interfaces with the pins via a serial boundary scan assembly including boundary scan **register** 1161 which allows all logic states at the actual pins of device 11 to be read or written. JTAG TAP controller 1151 and JTAG instruction **register** IR 1153 are provided on-chip. Test control 1205 and MPSD control 1203 are integrated into the circuitry . MPSD control. . .

DETD . . . as the following five distinct clocking domains in order to control domain data transfers with the scan clock (JCLK) and **application execution** with the functional clock (FCLK).

DETD . . . data is scanned to and from the device 11 through internal scan paths that are selected through a JTAG instruction **register** 1153. A unique JTAG opcode for each path allows entry of and access to internal scan data.

DETD FIG. 11 shows a further diagrammatic respective of the registers of FIG. 7 wherein JTAG instruction **register** IR 1153 is selected for scan between the terminals TDI and TDO. The IR 1153 is decoded in FIG.

```
       7.  .  .  requested by the control card 1141 of FIG. 2 via the serial
       line 1103. These shift registers are the bypass register 1167,
       the boundary scan register 1161, the message peripheral 1216
       of FIG. 9, the emulation control register 1251 and a pair of
       MPSD scan paths 1252 in the various domains and modules in the domains.
DETD   .  .  .  when not scanning and a zero when scanning. C0 and CX are
       sourced from the emulation control block adapter 1203 register
       1251. When host computer 1101 detects Ready to scan for all unlocked
       domains for a designated device on the target.  .  .
DETD   .  .  .  mode conditioned stop logic circuitry 1309B, 1309C and 1309A
in
       the domains respectively. The modes are established by a mode
     register 1311 which is scanable in FIGS. 11 and 14 to establish
       the type of stop and any other desired mode.  .  .
DETD   .  .  .  sent to the CPU domain 1213 to make it stop according to the
       stop mode established for it in mode register 1311 and mode
       conditioned stop circuitry 1309C. DONE circuitry 1363 detects when the
       stop is completed and signals back to.  .  .
DETD   .  .  .  an example of process steps by which the scan control 1149
       including IR 1153, adapter 1203 including ECR (emulation control
     register) 1251, and host computer 1101 cooperate to enter and
       perform sequences of commands on-chip.
DETD   .  .  .  to reach the state "Select-DR-Scan". This means that the scan
       controller is ready to accept scan into
<------User Break------>
u
=>
=> d 9


L6   ANSWER 9 OF 32  USPATFULL
AN     2000:25940  USPATFULL
TI     Processor condition sensing circuits, systems and methods
IN     Swoboda, Gary L., Sugar Land, TX, United States
       Ehlig, Peter N., Houston, TX, United States
PA     Texas Instruments Incorporated, Dallas, TX, United States (U.S.
       corporation)
PI     US 6032268  20000229
AI     US 1992-832661  19920204 (7)
RLI    Continuation of Ser. No. US 1989-388286, filed on 31 Jul 1989, now
       abandoned And a continuation-in-part of Ser. No. US 1987-93463, filed
on
       4 Sep 1987, now abandoned Ser. No. Ser. No. US 1987-140055, filed on 31
       Dec 1987, now patented, Pat. No. US 5109494 Ser. No. Ser. No. US
       1987-140192, filed on 31 Dec 1987, now patented, Pat. No. US 5101498
       Ser. No. Ser. No. US 1989-347968, filed on 4 May 1989, now abandoned
       Ser. No. Ser. No. US 1989-347969, filed on 4 May 1989, now abandoned
       Ser. No. Ser. No. US 1989-347605, filed on 4 May 1989, now abandoned
       Ser. No. Ser. No. US 1989-347596, filed on 4 May 1989, now patented,
       Pat. No. US 5072418 Ser. No. Ser. No. US 1989-347615, filed on 4 May
       1989, now patented, Pat. No. US 5142677 Ser. No. Ser. No. US
       1989-347966, filed on 4 May 1989, now patented, Pat. No. US 5155812 And
       Ser. No. US 1989-347967, filed on 4 May 1989, now abandoned
DT     Utility
LN.CNT 3972
INCL   INCLM: 714/030.000
       INCLS: 714/034.000; 714/726.000
NCL    NCLM:  714/030.000
       NCLS:  714/034.000; 714/726.000
IC     [7]
       ICM: G01R031-3177
       ICS: G01R031-3185
EXF    371/16.2; 371/19; 371/22.3; 371/22.5; 714/30; 714/34; 714/726
```

```
=> d ab 10


L6   ANSWER 10 OF 32  USPATFULL
AB     An apparatus and method for accelerating interpreters, interpretive
       environments, may manage pinning of a processor cache closest to a
       processor. An instruction set implementing a virtual machine may store
       each instruction in a single cache line as a compiled, linked loaded
       image. After loading, the cache is pinned, disabled from flushing the
       contents or replacing the contents of any cache line. A fast load may
       flush the cache and run an application containing the entire virtual
       machine instruction set. A pin manager may be hooked into a scheduler
in
       a multi-tasking operating system to load, pin, and unpin the processor
       cache as rapidly as needed. Thus, the processor cache may be available
       for general use, except when pinned for use by a virtual machine, such
       as an interpretive environment. Level-1 caches integrated into central
       processing units, particularly instruction caches or code caches are
       ideally suited to implementation of the invention.


=> d ab 11


L6   ANSWER 11 OF 32  USPATFULL
AB     A customizable data acquisition device (DA) and a data collection
system
       (DCS). The DA is remotely located for attachment to a variety of I/O
       devices, and uses an RF transceiver with a primary cache. A centrally
       located or mobile DCS with a secondary cache communicates with one or
       more DA's and stores DA data in the secondary cache. The DA has a
       programmable controller and an I/O interface capable of attaching to a
       variety of I/O devices. The DA primary cache holds data for bulk
       transfers to the DCS to minimize DA/DCS data transmission traffic. The
       DCS secondary cache accumulates data from the DA and allows data
queries
       by multiple independent application programs (which may be located on
       the DCS, other DCSs, or remote host systems) without requiring
       retransmission of data from the DA for each query. Applications can
       query cached data in the secondary cache, and optionally query the
       primary cache or query the I/O device via the DA. Each DA is capable of
       peer to peer communication to allow DAs physically located outside of
       the transmission range of the DA/DCS to communicate with the DCS by
       routing data through other DAs in a network of DAs until the data can
be
       transmitted to the DCS. I/O devices and DAs are shared by multiple
       applications in a hierarchical network of applications, host computers,
       DCSs, DAs, and I/O devices. A pass-through function allows the I/O
       interface to be emulated for software testing. When software testing is
       complete, the software is permanently stored in the DA.


=> d ab 12


L6   ANSWER 12 OF 32  USPATFULL
AB     A telephone instrument with a 12-key keypad that has relegendable
       buttons. The telephone instrument is coupled to apparatus such as a
       personal computer, network server, or switch that is capable of
       performing telephony functions. The telephony apparatus responds to an
       input from the keypad by providing a new set of legends for the keys
       and/or performing a telephone function. In a preferred embodiment, the
       telephone instrument is connected both to a personal computer and to a
       POTS telephone circuit and includes a separate keypad for the POTS
       telephone circuit. The apparatus includes a component which detects a
```

```
      failure in the personal computer and automatically deactivates the
      keypad with the relegendable buttons, leaving the telephone instrument
      still available for use with the POTS circuit. Also disclosed are a
      reduced-cost implementation of the telephone instrument and alternative
      embodiments in which the telephone instrument is connected to the
      telephony apparatus by a packet interface or itself contains the
      component which performs the telephony functions. The telephony
      functions may be those employed in switched circuit telephone systems
or
      those employed in packet telephone systems, and the systems themselves
      may be either wired or wireless.


=> d ab 13


L6    ANSWER 13 OF 32  USPATFULL
AB      A method for debugging a target application comprising Java code having
        native method dll's associated therewith. The method is carried out in
a
        computer having an operating system, a system debug application
        programming interface (API), and a Java virtual machine having a Java
        debug API. According to the method, the Java virtual machine is first
        launched under the system debug API. The application is then run under
        the Java virtual machine. Because the Java virtual machine that runs
the
        target application itself runs under the system debug API, simultaneous
        control of the target application via the system debug API and the Java
        debug API is enabled. Thus, the method effects the debug of the target
        application by simultaneously debugging the Java code and the native
        method dynamic load libraries.


=> d 13


L6    ANSWER 13 OF 32  USPATFULL
AN      1999:54524  USPATFULL
TI      Method for debugging a Java application having native method dynamic
        load libraries
IN      Edwards, Jonathan W., Lexington, KY, United States
        Evans, David H., Lexington, KY, United States
        Vassil, Michael G., Lexington, KY, United States
PA      International Business Machines Corporation, Armonk, NY, United States
        (U.S. corporation)
PI      US 5901315  19990504
AI      US 1997-874644  19970613 (8)
DT      Utility
LN.CNT 683
INCL    INCLM: 395/704.000
        INCLS: 395/701.000; 395/183.140; 395/183.010; 395/406.000; 395/500.000;
               395/527.000; 395/682.000
NCL     NCLM:  395/704.000
        NCLS:  395/500.430; 395/527.000; 395/701.000; 709/001.000; 709/302.000;
               709/305.000
IC      [6]
        ICM: G06F009-44
        ICS: G06F011-00
EXF     395/704; 395/701; 395/183.14; 395/183.13; 395/183.1; 395/183.01;
        395/185.01; 395/682; 395/500; 395/650; 395/406; 395/527


=> d ab 14
```

```
L6   ANSWER 14 OF 32  USPATFULL
AB     An apparatus and method for accelerating interpreters, interpretive
       environments, and the like optimizes the use of caches closest to a
       processor. An instruction set implementing a virtual machine
       (interpreter, interpretive environment) is written to fit each
       instruction at an individual cache line's address in the processor
       cache. The processor cache may be loaded with the instruction set in a
       compiled, linked, loaded image. After loading the processor cache, the
       cache is pinned, locked, disabled from flushing the contents or
       replacing the contents of any cache line. Faster loading of the
       processor cache may be achieved by flushing the processor cache and
       running an application containing all of the instructions of the
virtual
       machine instruction set. Level-1 processor caches integrated into
       central processing units, particularly instruction caches or code
caches
       are ideally suited to implementation of the invention. Examples include
       Intel's Pentium.TM. class products and Motorola's Power PC Processors.


=> d ab 15


L6   ANSWER 15 OF 32  USPATFULL
AB     A method for testing a digital processor 11 in which a test port 1149
is

       used to transfer trace data from the digital processor to a test host
       processor 1101 under control of a user definable program which
     executes in response to predetermined events on the digital
       processor. Trace data is gathered while an application program
       loaded in program memory 61 is executed by the digital
       processor. Trace data is temporarily stored in a trace region 99 of
data
       memory 25 by user definable code which is executed in a
       background manner by the digital processor in response to trigger
       events. The trigger events are also enabled by user definable code
which
       enables various portions of analysis hardware 1217. Trace data is
       transferred from the digital processor to the test host processor
       through test port 1149 by sending a notification signal to the test
host
       processor by means of message passing register 1216. The
       digital processor then monitors the message passing register
       for a handshake signal from the test host processor. When a handshake
       signal is received, trace data is written into the message passing
     register by user definable code in a background manner and
       transferred to the test host processor.


=> d ab 16


L6   ANSWER 16 OF 32  USPATFULL
AB     An emulation device (11) distributes common control information (8801)
       to each of a plurality of clock domains (1213, 1215, 1217) into which
       the emulation device is partitioned, and also provides the clock
domains
       with individualized clock control (8905, 8907, 8913).


=> d ab 17


L6   ANSWER 17 OF 32  USPATFULL
AB     A system and method for performing an emulation context switch save and
```

```
         restore in a processor that executes host applications and
         emulates guest applications. The processor includes an operating system
         and a first register that is saved and restored by the
         operating system during a host application context switch. The
         method and system comprises renaming the special-purpose
       register to the first register when emulating guest
         applications. When an emulation context switch occurs, a context save
         and restore of the special-purpose register is performed
         through the first register without operating system
         modification.


=> d kwic17

'KWIC17' IS NOT A VALID FORMAT
In a multifile environment, a format can only be used if it is valid
in at least one of the files.  Refer to file specific help messages
or the STNGUIDE file for information on formats available in
individual files.
REENTER DISPLAY FORMAT FOR ALL FILES (FILEDEFAULT):end


=> d kwic 17


L6   ANSWER 17 OF 32  USPATFULL
AB      A system and method for performing an emulation context switch save and
        restore in a processor that executes host applications and
        emulates guest applications. The processor includes an operating system
        and a first register that is saved and restored by the
        operating system during a host application context switch. The
        method and system comprises renaming the special-purpose
      register to the first register when emulating guest
        applications. When an emulation context switch occurs, a context save
        and restore of the special-purpose register is performed
        through the first register without operating system
        modification.
SUMM    The assignee of the present application has developed
        hardware that assists software emulation methods in order to
        obtain a solution that has the positive characteristics of both
methods:
        the small size of the interpreter and the speed of execution
        of the dynamic translator. Before the assisting hardware is used, the
        sequence of host instructions necessary to perform the function.  .  .

SUMM    .  .  .  This is referred to as a context save. When the process is
        subsequently swapped-back in, the operating system restores the
      register contents. This is referred to as a context restore.
SUMM    The present invention is method and system for performing an emulation
        context switch save and restore in a processor that executes
        host applications and emulates guest applications. The processor
        includes an operating system and a first register that is
        saved and restored by the operating system during a host
      application context switch. The method and system comprises
        renaming the special-purpose register to the first
      register when emulating guest applications. When an emulation
        context switch occurs, a context save and restore of the
special-purpose
      register is performed through the first register
        without operating system modification.
DETD    .  .  .  such as multiply, add, divide, and multiply-add. The FPRs 102
        are used to store the values generated by a floating-point
      application as the application executes.
DETD    When a floating-point application is swapped-out during a
        context switch, the operating system 104 performs a context save on
```

those FPRs 102 used by the **application** as follows. When another **application** is swapped-in that subsequently **executes** a floating-point instruction, the floating-point instruction causes what is known as a floating-point not available (FPNA) exception. The FPNA exception is an interrupt that causes the operating system 104 to save the contents of the previous **application** and to reset the FPNA so that the new **application** can use the FPRs 102.

CLM What is claimed is:

1. A method for performing emulation context switch save and restore in a processor that **executes** host applications and emulates guest applications, the processor including an operating system and a first **register** that is saved and restored by the operating system during a host **application** context switch, the method comprising the step of: (a) renaming a special-purpose **register** to the first **register** when emulating guest applications, wherein a floating-point **register** is provided as the first **register**; and (b) performing an emulation context switch, wherein a context save and restore of the special-purpose **register** is performed through the first **register** without operating system modification.

. . . as in claim 2 wherein step (a) further includes the step of: (a2) associating a direction bit with the special-purpose **register** and the floating-point **register** to indicate whether the special-purpose **register** or the floating-point **register** contains the most current value.

. . . claim 3 wherein step (b) further includes the step of: (b2) setting the direction bit to indicate that the special-purpose **register** contains the most current value.

. . . claim 4 wherein step (b) further includes the step of: (b3) resetting the direction bit to indicate that the floating-point **register** contains the most current value.

. . . step (b) further includes the step of: (b4) when performing an emulation context store, storing the contents of the special-purpose **register** if the direction bit is set, and storing the contents of the floating-point **register** if the direction bit is reset.

. . . claim 6 wherein step (b) further includes the steps of: (b4) when performing an emulation context restore, loading the floating-point **register** and the special-purpose **register**; and (b5) resetting the direction bit.

8. A method as in claim 7 wherein a pair of special-purpose registers is renamed to the floating-point **register**.

9. A processor that executes host applications and emulates guest applications, the processor comprising: an operating system; a first **register** that is saved and restored by the operating system during a host application context switch; a special-purpose **register**; means for renaming the special-purpose **register** to the first **register** when emulating guest applications, wherein the first **register** is a floating-point **register**; and means for performing an emulation context save and restore of the special-purpose **register** through the first **register** without operating system modification, wherein the means for performing an emulation context save and restore of the special-purpose **register** are user-level instructions.

11. A processor as in claim 10 further including a direction bit associated with the special-purpose **register** and the

```
     floating-point register to indicate whether the
     special-purpose register or the floating-point
   register contains the most current value.

     12. A processor as in claim 11 wherein the direction bit is set to
     indicate that the special-purpose register contains the most
     current value.

     13. A processor as in claim 12 wherein the direction bit is reset to
     indicate that the floating-point register contains the most
     current value.

 . . .  context store is performed, if the direction bit is set, then the
     operating system stores the contents of the special-purpose
   register, and if the direction bit is reset, then the operating
     system stores the contents of the floating-point register.

 . . .  A processor as in claim 14 wherein when an emulation context restore
     is performed, the operating system loads the floating-point
   register and the special-purpose register, and resets
     the direction bit.

     16. A processor as in claim 15 wherein a pair of special-purpose
     registers is renamed to the floating-point register.

     17. A processor for executing host applications and emulating guest
     applications, each of the guest applications including guest
     instructions, the.  . .  a plurality of floating-point registers for
     containing values, wherein each pair of special-purpose registers are
     renamed to a particular floating-point register; and a
     direction bit associated with each pair of special-purpose registers
and
     the corresponding floating-point register to indicate whether
     the pair of special-purpose registers or the corresponding
     floating-point register contains the most current value;
     wherein the operating system is responsive to the direction bit to
     perform an emulation context.  . .


=> d 17


L6   ANSWER 17 OF 32  USPATFULL
AN      1998:117089  USPATFULL
TI      Method and system for performing an emulation context save and restore
        that is transparent to the operating system
IN      Kahle, James Allan, Austin, TX, United States
        Mallick, Soummya, Austin, TX, United States
        Martin-de-Nicolas, Arturo, Austin, TX, United States
PA      International Business Machines Corporation, Armonk, NY, United States
        (U.S. corporation)
PI      US 5812823  19980922
AI      US 5817943  19960102 (8)
DT      Utility
LN.CNT 391
INCL    INCLM: 395/500.000
        INCLS: 395/563.000; 395/569.000; 395/570.000
NCL     NCLM:  395/500.470
        NCLS:  395/500.480; 712/222.000; 712/228.000; 712/229.000
IC      [6]
        ICM: G06F009-455
EXF     395/500; 395/563; 395/567; 395/569; 395/570; 395/527; 395/385; 395/580;
        395/584
```

```
=> d ab 18


L6   ANSWER 18 OF 32  USPATFULL
AB     An electronic device having addressable storage elements and a bus so
       that the storage elements are accessible via the bus, an address
     register connected to the bus, a data register
       connected to the bus, terminals for serial scan-in and scan-out, a
       scanable emulation control register coupled to the terminals,
       and a selecting circuit responsive to bits in the emulation control
     register for coupling the address register and the
       data register to the terminals to enable scanning of the
       address and data registers.


=> d ab 19


L6   ANSWER 19 OF 32  USPATFULL
AB     Telephone station equipment consisting of a phone device interconnected
       with a personal computer. The phone device includes a conventional
       telephone handset and a keypad employing pushbutton display keys each
of
       which has a writable keyface display for visually indicating the
       function of the key or other information to the user. The personal
       computer is connected to both the phone device and to one or more
       telephone communications channels and is programmed to display
prompting
       information on the key displays and respond to keypress events to
       perform the functions indicated. The user can perform a variety of
       telephone system management tasks solely by viewing and manipulating
the
       phone device keypad, including manual dialing, redialing, speed-dialing
       from a directory of commonly called numbers, making flash
       disconnections, forwaring calls, controlling call waiting and caller
       I.D. functions, adjusting speakerphone volume and microphone gain,
       handling conference calls, automatically logging into remote databases,
       recording the time and nature of each call in an accounting file,
       performing unattended call answering and voice mail functions,
utilizing
       voice responsive and automated voice output systems, and other
telephone
       management functions.


=> d ab 20


L6   ANSWER 20 OF 32  USPATFULL
AB     A system and method for improving the performance of a processor that
       emulates a guest instruction where the guest instruction includes a
       first and second operand. The first operand is stored in a general
       purpose register, and the second operand is stored in a
       special-purpose register. The method and system provides a
       host instruction that performs an operation using the first operand and
       the second operand without moving the second operand from the
       special-purpose register into the general purpose
     register. This reduces the number of instructions in the
       semantic routines necessary to operate on immediate data from guest
       instructions and increases emulation performance.


=> d ab 21
```

```
L6    ANSWER 21 OF 32  USPATFULL
AB       What is disclosed is an emulator that emulates on an execution machine
         the operation of a target machine. The emulator emulates routines that
         are called via a jump table such as the BIOS. Control is transferred to
         an emulation module not by directly trapping the procedure to call the
         BIOS but by placing a privileged instruction (a halt instruction, for
         example) in the area called and by causing a trap through the execution
         of the privileged instruction. An identifier is placed after the halt
         instruction and a needed BIOS emulation module is called by a
dispatcher
         using this identifier. Therefore, normal operation can be obtained even
         if there is a resident program which rewrites the jump table.


=> d 21


L6    ANSWER 21 OF 32  USPATFULL
AN       1998:59535  USPATFULL
TI       Computer emulator
IN       Ogata, Hideaki, Nagano-Ken, Japan
         Tanimoto, Akihito, Nagano-Ken, Japan
         Nakaoka, Yasushi, Nagano-Ken, Japan
         Kojima, Masanori, Nagano-Ken, Japan
         Akahori, Yutaka, Nagano-Ken, Japan
PA       Seiko Epson Corporation, Tokyo, Japan (non-U.S. corporation)
PI       US 5758124  19980526
AI       US 1995-559223  19951117 (8)
PRAI     JP 1994-309813      19941118
DT       Utility
LN.CNT 883
INCL     INCLM: 395/500.000
         INCLS: 395/527.000; 395/421.030; 364/DIG.001; 364/232.300; 364/DIG.002;
                364/927.810; 364/955.500
NCL      NCLM:  395/500.480
         NCLS:  395/527.000; 711/213.000
IC       [6]
         ICM: G06F009-455
EXF      395/500; 395/375; 395/750; 395/800; 395/416; 395/527; 395/421.03;
         364/200MSFile; 364/900MSFile; 364/927.81


=> d ab 22


L6    ANSWER 22 OF 32  USPATFULL
AB       Application programs compiled for a first, "source", computer
         are translated, from their object form, for execution on a
         second, "target", computer. The translated application
         programs are linked or otherwise bound with a translation of the source
         computer system software. The translated system software operates on
the
         image of the source computer address space in the target computer
         exactly as it did in the source computer. The semantics of the source
         computer system software are thus preserved identically. In addition, a
         virtual hardware environment is provided in the target computer to
         manage events and to deal with differences in the address space layouts
         between the source and target computers.


=> d ab 23


L6    ANSWER 23 OF 32  USPATFULL
AB       A method and system provides a special purpose or embedded system
```

```
       developer with the ability to confirm the correct operation of a
       computer program designed to operate on a target system whose
processing
       and storage capabilities may be more austere than the host system upon
       which the computer program is designed and tested. A key feature of the
       method and the system enables a developer to execute and debug
       an application program on a host system while observing and
       testing the operation of the program through the input/output of the
       target system. Another feature of the method and system is an
     application loader that dynamically sizes and, as necessary,
       reconfigures the available memory to permit multiple applications to
       reside simultaneously on the target system by resolving addresses in
the
       target system at the time an application of interest is
       downloaded to the target system.


=> d 23


L6   ANSWER 23 OF 32  USPATFULL
AN     1998:12908  USPATFULL
TI     Method and system for loading and confirming correct operation of an
       application program in a target system
IN     Barnstijn, Michael A., Kitchener, Canada
       Church, Mark E., Kitchener, Canada
       Linkert, Barry W., Kitchener, Canada
       Lazaridis, Mihal, Waterloo, Canada
PA     Research In Motion Limited, Waterloo, Canada (non-U.S. corporation)
PI     US 5715387  19980203
AI     US 1996-742632  19961101 (8)
RLI    Continuation of Ser. No. US 1995-386528, filed on 10 Feb 1995, now
       patented, Pat. No. US 5600790
DT     Utility
LN.CNT 746
INCL   INCLM: 395/183.140
       INCLS: 395/183.180; 395/704.000
NCL    NCLM:  714/038.000
       NCLS:  395/704.000; 714/042.000
IC     [6]
       ICM: G06F009-455
EXF    395/183.14; 395/705; 395/183.18; 395/704; 395/427; 395/497.02


=> d ab 24


L6   ANSWER 24 OF 32  USPATFULL
AB     A Target MCU is restored to a Target State. A Host Trace of Debug
       Commands is preserved as the Target MCU is driven from a known first
       state to the Target State by executing a series of Debug Commands. The
       Target MCU is then reinitialized to the known first state. The Debug
       Commands are read from the Host Trace and sent to a Modular Development
       System (MDS) for execution by the Target MCU until the Target MCU is
       again is driven to the Target State.


=> d kwic 24


L6   ANSWER 24 OF 32  USPATFULL
SUMM   . . . "real-time debugger" in that they cover half of the bases. A
       developer must not only be able to observe the execution of
       his application in real-time, but also be able to change
       parameters of his application, and be able to generate events
```

in real-time order to modify program flow-control. This functionality would give the developer a better feel for the **execution** of his **application**.

DETD The MCUdebug software system is an **application development** and debugging environment for embedded microcontrollers. It allows users to edit, build, and debug their applications all within the same graphical user interface. It dynamically supports a variety of **execution** targets, including in-circuit **emulators**, evaluation systems and boards, and instruction set simulators.

DETD . . . on a different platform (such as a personal computer or a workstation), and then downloaded into the MCU memory for **execution** and debugging. For example, one could develop his **application** for a Motorola MC68HC08.TM. microcontroller using tools available for either Microsoft Windows.TM. or UNIX.TM.. Using a debugger, the object code generated can then be downloaded into the target evaluation system for **executing** and debugging the **application**.

DETD Since embedded MCUs do not support basic input/output, a console window is not available where a developer can observe the **execution** of his **application**. Also, no interaction can take place between an **application** and its developer during program **execution**. Normally, a debugger can be used to provide something close to a console, since it allows a user to observe the state of his **application** at any point. Debuggers also allow a user to perform traditional debugging operations like instruction tracing and programmed breakpoints.

DETD . . . in embedded systems, since these capabilities allow a user the capability to track the exact flow of control through his **application**. These capabilities can be used without stopping program **execution**, and thus are particularly useful in real-time systems for which breakpointing would be inconvenient.

DETD . . . command line within the MCUdebug environment, a user has to perform the following: in the "Customize Assembler" dialog, the "Assembler **Executable**" box should be set to "casm05, and "Assembler Options" box should be set to the string "l s d". No further setup is required for this **application** (since it only requires an assembler). If the toolset contained a compiler, assembler, linker, and locator, a little more toolset. . .

DETD **Execution** control is essential to observe the flow of **execution** through a program. MCUdebug provides the user a wide variety of commands to support **execution** control. Commands such as "animate", "go", "gotil", "trace", and "step" are provided so a user can run his **application**, trace instructions, or **execute** his **application** until a specific section is reached.

DETD The MCU reset vector is used extensively in embedded applications to return an **application** to a known starting state in case of abnormal program **execution**. It can also be used by a user externally while debugging to return his **application** to a known starting state. The MCUdebug command "reset" loads the MCU program counter ("pc") with the contents of the. . . the "resetgo" command will load the MCU program counter ("pc") with the contents of the reset vector and begin program **execution**.

DETD For example, to run his **application** a user can use a "go" command. That starts **execution** of his **application** program, which runs until the user types a "stop" command. If the user wishes to **execute** his **application** until a certain address, he can use a "gotil" command (he must specify the ending address). If the user wishes to trace the **execution** of a single instruction, he can use a "t" command (an additional parameter <n> can be specified which allows them to trace <n> instructions). If

the user wish to return to the starting poin f the
application, he can use a "reset" command.

DETD Script, command, or log files allow a user to maintain a log of all the commands that he has **executed** during an MCUdebug session (command names "script" or "cf"). It also allows the user the capability to perform regression tests on his target embedded **application** . These scripts are ASCII files, so a user can copy or edit these files for reuse, as necessary.

DETD . . . to display a source window with a highlight line (the highlight line reflects the exact point in the source). Visual **execution** control allows a user a more abstract mechanism for controlling program **execution**. Instead of keeping track of explicit addresses in his **application**, he can control program **execution** using actual source code.

```
DETD                                         . . .  EQU  $40
PA5.    EQU  $20
PA4.    EQU  $10
PA3.    EQU  $08
PA2.    EQU  $04
PA1.    EQU  $02
PA0.    EQU  $01
DDRA    EQU  $04       ;PORT A DATA DIRECTION REGISTER
DDRA7   EQU  7         ;BIT #7 OF PORT A DDR
DDRA6   EQU  6
DDRA5   EQU  5
DDRA4   EQU  4
DDRA3   EQU  3
DDRA2   EQU  2
DDRA1   EQU  1
DDRA0.   .   .  OF DDRA7
DDRA6   EQU  $40
DDRA5.  EQU  $20
DDRA4.  EQU  $10
DDRA3.  EQU  $08
DDRA2.  EQU  $04
DDRA1.  EQU  $02
DDRA0.  EQU  $01
TCR     EQU  $12       ;TIMER CONTROL REGISTER
ICIE    EQU  7         ;INPUT CAPTURE INTERRUPT ENABLE BIT
OCIE    EQU  6         ;OUTPUT COMPARE INTERRUPT ENABLE BIT
TOIE                   ;TIMER OVERFLOW INTERRUPT ENABLE BIT
ICIE.   EQU.  .   .   BIT POSITION
OCIE.   EQU  $40       ;OUTPUT COMPARE INTERRUPT BIT POSITION
TOIE.   EQU  $20       ;TIMER OVERFLOW INTERRUPT BIT POSITION
TSR     EQU  $13       ;TIMER STATUS REGISTER
ICF     EQU  7         ;INPUT CAPTURE FLAG BIT
OCF     EQU  6         ;OUTPUT COMPARE FLAG BIT
TOF     EQU  5         ;TIMER OVERFLOW FLAG BIT
ICF.    EQU  $80.  .   .   ;INPUT CAPTURE FLAG BIT POSITION
OCF.    EQU  $40       ;OUTPUT COMPARE BIT POSITION
TOF.    EQU  $20       ;TIMER BIT POSITION
OCRH    EQU  $16       ;OUTPUT COMPARE REGISTER (HIGH BYTE)
OCRL    EQU  $17       ;OUTPUT COMPARE REG (LOW BYTE)
TCRH    EQU  $18       ;TIMER COUNTER REGISTER (HIGH BYTE)
TCRL    EQU  $19       ;TIMER COUNTER REG (LOW BYTE)
ACRH    EQU  $1A       ;ALTERNATE COUNTER REG (HIGH BYTE)
ACRL    EQU  $1B       ;ALTERNATE COUNTER.  .   .  ;START OF ON-CHIP RAM
ROMStart
        EQU  $0100     ;START OF ON-CHIP ROM
ROMEnd  EQU  $08FF     ;END OF ON-CHIP ROM
Vectors
        EQU  $1FF8     ;RESET/INTERRUPT VECTOR AREA
```

Application specific equates

```
LED     EQU   PA7         D ON WHEN PA7 IS LOW
LED.    EQU   PA7.       ;ED BIT POSITION
Put program variables here (Use RMBs)
.  .  .  INTERRUPTS
        STA   TCR
        BRCLR OCF,TSR,NEXT ;IF OCF IS NOT SET, SKIP INTERRUPT FLAG
RESET
        LDA   TSR      ;READ TIMER STATUS & OUTPUT COMPARE
REGISTER
        LDA   OCRL     ;TO CLEAR THE INTERRUPT
NEXT    LDA   #3       ;OCFs COUNT 3.fwdarw.0
        STA   OCFs     ;SET OUTPUT COMPARE INTERRUPT COUNT
TO 3
        CLR.  .  .  ;SET OCR SO THAT INTERRUPT IS GENERATED
             ;ON THE NEXT OCCURENCE OF THE COUNT VALUE
             ;COPIED FROM THE TIMER COUNTER REGISTER.
             ;THIS COUNT WILL OCCUR AFTER ONE COMPLETE
             ;CYCLE OF THE COUNTER, THAT IS 262,144 CYCLES
             ;FOR A TOTAL OF 131.072ms.  .  .  INT
             ;OPERATING FREQUENCY)
        CLI  ;CLEAR I BIT IN STATUS REG TO ENABLE INTERRUPT
MAIN -- Beginning of main program loop. Loop is
executed once every 400ms (393.216ms). A pass through
all major task routinestakes less than 400mS and then
time is wasted until MSB of.  .  .  interrupt, the
interrupt is serviced (OCFs gets decremented (3.fwdarw.0)
and then cleared by reading the TSR and low byte of
the Output Compare Register (OCRL). When OCFs=0, MSB
of TIC gets set and OCFs is set back to 3.
(3*131.072ms/OCF = 393.216ms)
The variable TIC keeps track.  .  .


=> d kwic 25


L6   ANSWER 25 OF 32  USPATFULL
SUMM   In addition to testing for functionality and for manufacturing defects,
     application software development requires a similar
       level of simulation, observability and controllability in the system or
       sub-system design phase. The emulation phase of design should
       ensure that an IC (integrated circuit), or set of ICs, functions
       correctly in the end equipment.  .  .
SUMM   .  .  .  simulation and continually increasing cost of CAD (computer
       aided design) tools. In the board design the side effects include
       decreased register visibility and control, complicated debug
       and simulation in design verification, loss of conventional
     emulation due to loss of physical access by packaging many
       circuits in one package, increased routing complexity on the board,
       increased costs of design tools, mixed-mode packaging, and design for
       produceability. In application development, some
       side effects are decreased visibility of states, high speed
     emulation difficulties, scaled time simulation, increased
       debugging complexity, and increased costs of emulators.
       Production side effects involve decreased visibility and control,
       complications in test vectors and models, increased test complexity,
       mixed-mode packaging, continually.  .  .
SUMM   Among the objects of the present invention are to provide improved
     emulation, simulation and testability architectures and methods
       which provide visibility and control without physical probing or
special
       test fixtures; to provide improved emulation, simulation and
       testability architectures and methods which are applicable to critical
       components of system designs to support test and integration of both
       hardware and software; to provide improved emulation,
        simulation and testability architectures and methods that are a viable
        alternative to high capital-cost test equipment and systems; to provide
        improved emulation, simulation and testability architectures
        and methods which integrate access to sophisticated operations in
        hardware emulation, fault emulation, simulation and
        built-in test; to provide improved emulation, simulation and
        testability architectures and methods which apply hardware and software
        visibility and control to reduce application
      development time and thus reduce the user manufacturer's
        time-to-market on new products; and to provide improved
      emulation, simulation and testability architectures and methods
        to leverage hierarchical partitioning and automatically generate
        reusable tests for related chips and systems.
SUMM    . . . one form of the invention is an electronic system including
        electronic circuitry to be tested and including serial scan shift
      register latches, and a serial scan generator embedded in the
        system in manufacture and connected to the electronic circuitry thereby
        facilitating. . .
DETD    . . . 347,967 (TI-14145), and Ser. No. 347,969 (TI-14147). For
        example, device 11 has sets of serially connected set of serial shift
      register latches (SRLS) distributed through the device like
        beads on a string for input and output of emulation signals serially.
        The. . .
DETD    . . . additions is simply connected by a parallel cable 304 to
        terminals 1189 to parallel bus 1187 to SBM-2 on the application
        system 1181 in the field, or even in manufacture. A test program is
        inserted in host computer 300 and executed.
DETD    In read synchronization, circuitry in SBM 101 supports an Update
command
        that loads all status register***
<------User Break------>
u
=>
=> d ab 25


L6   ANSWER 25 OF 32  USPATFULL
AB      An electronic system for use with a host computer. The system includes
        electronic circuitry including a first semiconductor chip generally
        operable for a first function and also adapted for input and output of
        emulation signals. This is combined with emulation circuitry including
a

        second semiconductor chip adapted for connection to the host computer.
        The emulation circuitry is connected to the electronic circuitry to
        generate emulation signals to input to the electronic circuitry and to
        accept emulation signals from the electronic circuitry. A physical
        assembly supports the emulation circuitry and the electronic circuitry
        as a unit. Other electronic systems and emulation and testing devices,
        cables, systems and methods are also disclosed.


=> d 25


L6   ANSWER 25 OF 32  USPATFULL
AN      97:102464  USPATFULL
TI      Electronic systems and emulation and testing devices, cables, systems
        and methods
IN      Swoboda, Gary L., Sugar Land, TX, United States
        Hoar, Henry R., Houston, TX, United States
        Coomes, Joseph A., Missouri City, TX, United States
PA      Texas Instruments Incorporated, Dallas, TX, United States (U.S.
        corporation)
PI      US 5684721  19971104
AI      US 1992-851232  19920313 (7)
```

```
RLI    Continuation of Ser. No. US 1989-401198, filed on 28 Aug 1989, now
       abandoned which is a continuation-in-part of Ser. No. US 1986-948337,
       filed on 3 Dec 1986, now abandoned which is a continuation-in-part of
       Ser. No. US 1987-93463, filed on 4 Sep 1987, now abandoned
DT     Utility
LN.CNT 2012
INCL   INCLM: 364/578.000
       INCLS: 364/488.000; 364/489.000; 395/500.000
NCL    NCLM:  395/500.440
IC     [6]
       ICM: G06F017-00
       ICS: G06G007-06
EXF    364/578; 364/488; 364/489; 371/23; 371/22.2; 371/22.3; 395/500


=> d ab 26


L6   ANSWER 26 OF 32  USPATFULL
AB      An emulation device (11) distributes common control information (8801)
        to each of a plurality of clock domains (1213, 1215, 1217) into which
        the emulation device is partitioned, and also provides the clock
domains
        with individualized clock control (8905, 8907, 8913).


=> d ab 27


L6   ANSWER 27 OF 32  USPATFULL
AB      A method and system provides a special purpose or embedded system
        developer with the ability to confirm the correct operation of a
        computer program designed to operate on a target system whose
processing
        and storage capabilities may be more austere than the host system upon
        which the computer program is designed and tested. A key feature of the
        method and the system enables a developer to execute and debug
        an application program on a host system while observing and
        testing the operation of the program through the input/output of the
        target system. Another feature of the method and system is an
      application loader that dynamically sizes and, as necessary,
        reconfigures the available memory to permit multiple applications to
        reside simultaneously on the target system by resolving addresses in
the
        target system at the time an application of interest is
        downloaded to the target system.


=> d 27


L6   ANSWER 27 OF 32  USPATFULL
AN      97:10989  USPATFULL
TI      Method and system for loading and confirming correct operation of an
        application program in a target system
IN      Barnstijn, Michael A., Kitchener, Canada
        Church, Mark E., Kitchener, Canada
        Linkert, Barry W., Kitchener, Canada
        Lazaridis, Mihal, Waterloo, Canada
PA      Research In Motion Limited, Waterloo, Canada (non-U.S. corporation)
PI      US 5600790  19970204
AI      US 1995-386528  19950210 (8)
DT      Utility
LN.CNT 798
INCL    INCLM: 395/183.140
```

```
       INCLS: 395/5[illegible]000; 395/200.010
NCL    NCLM:  714/03[illegible].000
       NCLS:  395/500.440
IC     [6]
       ICM: G06F009-455
EXF    395/183.14; 395/183.13; 395/183.01; 395/700; 395/650

=> d ab 28

L6   ANSWER 28 OF 32  USPATFULL
AB     An integrated circuit terminal of a data processing system (10) is used
       to communicate multiplexed signals with an external device. During a
       reset operation in which a reset signal is asserted, a desired internal
       clock signal is driven to the integrated circuit terminal such that an
       emulation system (52) may use the internal clock signal to synchronize
       an emulation operation. After the reset signal is negated, the
emulation
       system synthesizes the internal clock signal for use during emulation.
       External visibility of a write operation to a register which
       controls pertinent signal parameters is provided via other integrated
       circuit terminals when the data processor operates in an emulation
mode.
       The external visibility allows the development system to make similar
       changes to corresponding signal parameters therein. Therefore, the
       development system is able to accurately synchronize an emulation
       operation even when signal parameters are modified during operation.

=> d ab 29

L6   ANSWER 29 OF 32  USPATFULL
AB     Operations of a data processing device are traced by detecting a jump
       address in the program counter sequence, and pushing the jump address
       onto a trace stack.

=> d ab 30

L6   ANSWER 30 OF 32  USPATFULL
AB     An emulation device including a serial scan testability interface
having
       at least first and second scan paths, and state machine circuitry
       connected and responsive to said second scan path generally operable
for
       emulation control of logical circuitry associated with said emulation
       device.

=> d kwic 30

L6   ANSWER 30 OF 32  USPATFULL
SUMM   In addition to testing for functionality and for manufacturing defects,
     application software development requires a similar
       level of simulation, observability and controllability in the system or
       sub-system design phase. The emulation phase of design should
       ensure that an IC (integrated circuit), or set of ICs, functions
       correctly in the end equipment.  .  .
SUMM   .  .  .  simulation and continually increasing cost of CAD (computer
       aided design) tools. In the board design the side effects include
       decreased register visibility and control, complicated debug
```

and simulation in design verification, loss of conventional **emulation** due to loss of physical access by packaging many circuits in one package, increased routing complexity on the board, increased costs of design tools, mixed-mode packaging, and design for produceability. In **application development**, some side effects are decreased visibility of states, high speed **emulation** difficulties, scaled time simulation, increased debugging complexity, and increased costs of **emulators**. Production side effects involve decreased visibility and control, complications in test vectors and models, increased test complexity, mixed-mode packaging, continually. . .

SUMM Among the objects of the present invention are to provide improved **emulation**, simulation and testability architectures and methods which provide visibility and control without physical probing or special test fixtures; to provide improved **emulation**, simulation and testability architectures and methods which are applicable to critical components of system designs to support test and integration of both hardware and software; to provide improved **emulation**, simulation and testability architectures and methods that are a viable alternative to high capital-cost test equipment and systems; to provide improved **emulation**, simulation and testability architectures and methods which integrate access to sophisticated operations in hardware **emulation**, fault **emulation**, simulation and built-in tests to provide improved **emulation**, simulation and testability architectures and methods which apply hardware and software visibility and control to reduce **application development** time and thus reduce the user manufacturer's time-to-market on new products; and to provide improved **emulation**, simulation and testability architectures and methods to leverage hierarchical partitioning and automatically generate reusable tests for related chips and systems.

DRWD FIG. 59A is a compact diagram of shift **register** latches SRLs in a scan chain in FIG. 59;

DRWD FIG. 83 is a schematic of a parallel **register** latch for use in the GSP chip of FIG. 80;

DRWD FIG. 84 is a schematic of a serial **register** latch for use in the GSP chip of FIG. 80;

DETD Extended development system 1101 provides full-speed, in-circuit **emulation** for system design and for hardware and software debug on widely available personal computer systems. The development tools provide technological. . . to prototype. The development system elements provide ease of use and offer the designer the tools needed to significantly reduce **application** system **development** time and cost to put designs into production faster.

DETD . . . configuration file, a GSP (graphic signal processor) configuration, a programmable array logic (PAL) file, an ASIC file and a GPL **register** file.

DETD . . . software and hardware with the target system. An important emulation interface provides control and access to every memory location and **register** of the target chip and extend the device architecture as an attached processor.

DETD . . . before the target system is complete. To accomplish this, code is downloaded into the memory on the board 1043 and **executed** at full speed via the interface on an **application** board used in place of the incomplete target system. A suitable **application** board includes a DSP 11, 16K.times.32 bits of full-speed (zero wait states) SRAM on a primary bus, two selectable banks. . .

DETD . . . in coassigned application Ser. No. 057,078 issued Aug. 22, 1989, (U.S. Pat. No. 4,860,290) and incorporated herein by reference. Shift **register** latches (SRLs) designated "S" are distributed through the device 11 like a string of beads on a serial scan path. . .

DETD . . . TDO communicate with the system. TMS and TCK communicate with a
tap controller 1151 which is connected to an instruction
register 1153 and an instruction decoding circuit 1155.

DETD Test access port (TAP) controller 1151 is in turn coupled to instruction
register (IR) 1153 and a first multiplexer 1173. The instruction
register can receive serial scan signals from the TDI line and
output serially to MUX 1173. MUX 1173 is under control of the TAP and
can select the output signal from the instruction **register** or
from another MUX 1171.

DETD The instruction **register** also controls a bypass
register (BR) 1167 and one or more boundary scan registers (BSR)
1161. The bypass **register** receives the TDI signal and outputs
it to MUX 1171. MUX 1171 is under control of the instruction
register 1153. Based on the instruction loaded into the
instruction **register**, MUX 1171 outputs its input from the
bypass **register** or its input from one or more BSRs, or
internal device **register** scan. Each boundary scan
register is controlled via the test access port and the
instruction **register**.

DETD . . . or a test mode. During the normal mode, input data entering
terminals of IC logic passes through the boundary scan **register**
, into the IC logic and out to the normal output terminals without any
change due to the BSR. During the. . . test mode, normal input data
is interrupted, and test input data is captured, shifted, and updated
within the boundary scan **register**. The boundary scan
register includes two memories, a first memory for receiving and
shifting data from the TDI line and a second memory for. . .

DETD . . . controller card 1141 through pin TDI and enters any one of a
number of shift registers, including a boundary scan **register**
1161, a device identification **register** 1163 and design
specific test data registers 1165. A bypass **register** 1167 is
also provided. These shift registers or serial scan registers are
selected via a MUX 1171 under the control. . . output from MUX 1171
is fed to a MUX 1173 so that under control of tap controller 1151 the
instruction **register** 1153 or MUX 1171 is selected by MUX 1173.
JTAG clock TCK and MUX 1173 output are fed to flip. . .

DETD . . . has two scan paths. One of the scan paths is termed the MPSD
data path which usually has numerous shift **register** latches S
(or SRL) serially interconnected like a string of beads throughout the
module. The second scan path is termed the MPSD control path which
generally has fewer shift **register** latches and which selects
which MPSD data paths are to be scanned. These scan paths are described
in above-cited application. . .

DETD . . . core domain 1213, system domain 1215 and analysis domain 1217
are shown in FIG. 50 and interface through the shift **register**
latches of FIG. 47 to all of the various parts of the chip.

DETD . . . chip of device 11. JTAG control 1201 interfaces with the pins
via a serial boundary scan assembly including boundary scan
register 1161 which allows all logic states at the actual pins
of device 11 to be read or written. JTAG TAP controller 1151 and JTAG
instruction **register** IR 1153 are provided on-chip. Test
control 1205 and MPSD control 1203 are integrated into the circuitry.
MPSD control 1203. . .

DETD . . . as the following five distinct clocking domains in order to
control domain data transfers with the scan clock (JCLK) and
application execution with the functional clock
(FCLK).

DETD . . . data is scanned to and from the device 11 through internal scan
paths that are selected through a JTAG instruction **register**
1153. A unique JTAG opcode for each path allows entry of and access to
internal scan data.

DETD FIG. 54 shows a further diagrammatic perspective of the registers of

FIG. 50 where[illegible]TAG instruction **register** IR 1[illegible] is selected for scan betw[illegible] the terminals TDI and TDO. The[illegible] 1153 is decoded in FIG. 50. . . . requested by the control card 1141 of FIG. 45 via the serial line 1103. These shift registers are the bypass **register** 1167, the boundary scan **register** 1161, the message peripheral 1216 of FIG. 52, the emulation control **register** 1251 and a pair of MPSD scan paths 1252 in the various domains and modules in the domains.

DETD . . . when not scanning and a zero when scanning. CO and CX are sourced from the emulation control block adapter 1203 **register** 1251. When host computer 1101 detects Ready to Scan for all unlocked domains for a designated device on the target. . . .

DETD . . . mode conditioned stop logic circuitry 13098, 1309C and 1309A in the domains respectively. The modes are established by a mode **register** 1311 which is scanable in FIGS. 54 and 57 to establish the type of stop and any other desired mode. . . .

DETD . . . sent to the CPU domain 1213 to make it stop according to the stop mode established for it in mode **register** 1311 and mode conditioned stop circuitry 1309C. DONE circuitry 1363 detects when the stop is completed and signals back to. . . .

DETD . . . an example of process steps by which the scan control 1149 including IR 1153, adapter 1203 including ECR (emulation control **register**) 1251, and host computer 1101 cooperate to enter and perform sequences of commands on-chip.

DETD . . . to reach the state "Select-DR-Scan". This means that the scan controller is ready to accept scan into a DR (data **register**) identified just before as the ECR 1251 by the host to the IR 1153. Into ECR 1251 the host 1101. . . .

DETD In FIG. 59, serial scan bits enter the emulation control **register** ECR 1251 which is subdivided into a shift **register** LOCK 1351 for holding bits to lock and unlock domains, a first C0,CX control code shift **register** named CODA 1353, a second C0,CX control code shift **register** named CODB 1357, a shift **register** 1359 associated with event manager circuitry 1365, and a two-bit **register** JMODE 1360. These registers are compactly illustrated in FIG. 59A. Thus, the serial scan enters on a scan line SIN,. . . .

DETD A code state machine 1381 controls a two input MUX 1383. MUX 1383 selects the C0,CX two-bit contents of shift **register** 1353 or 1357 and loads them into an enabled one of three flip flops 1393, 1395 and 1397. A lock control circuit 1401 operating under the control of lock shift **register** 1351 and code state machine 1381 sends lock signals to disable or enable each of the flip flops 1393, 1395. . .

DETD . . . the test codes on line 1421, thus overriding the code state machine feature. This option is selected by scanning JMODE **register** with "00" (both bits zero). Thus, the preferred embodiment is accommodates direct host control of the domains, wherein the latter. . . .

DETD 2) Program state machine 1381 operations via REVT (**Register** Event) **register** 1359 to respond to stimuli including:

DETD . . . to JCLK (with a code 00 in both registers 1353 and 1357) and vice versa for each domain, via LOCK **register** 1351; and

DETD The operation of code state machine 1381 is now further described. When the JTAG IR (Instruction **Register** 1153) is loaded with a scan path select command for path 1251, a line ECRSEL feeds a signal to state. . . . whereupon the state machine 1381 enters a lock state. This allows the registers 1351, 1353 and 1357, the event manager **register** 1359, and JMODE **register** 1360 to be changed without disturbing the MPSD codes and clocks supplied from flip flops 1393, 1395, 1397 and selection circuits 1

<------User Break------>

u

```
=>
=> d 30


L6   ANSWER 30 OF 32  USPATFULL
AN     94:60720  USPATFULL
TI     Emulation devices, systems and methods utilizing state machines
IN     Swoboda, Gary L., Sugar Land, TX, United States
       Daniels, Martin D., Houston, TX, United States
       Coomes, Joseph A., Missouri City, TX, United States
PA     Texas Instruments Incorporated, Dallas, TX, United States (U.S.
       corporation)
PI     US 5329471  19940712
AI     US 1993-84787  19930629 (8)
RLI    Continuation of Ser. No. US 1992-911250, filed on 7 Jul 1992, now
       abandoned which is a continuation of Ser. No. US 1989-387549, filed on
       31 Jul 1989, now abandoned which is a continuation-in-part of Ser. No.
       US 1987-93463, filed on 4 Sep 1987, now abandoned And a
       continuation-in-part of Ser. No. US 1987-57078, filed on 2 Jun 1987,
now
       patented, Pat. No. US 4860290, issued on 2 Jun 1987
DT     Utility
LN.CNT 4145
INCL   INCLM: 364/578.000
       INCLS: 364/DIG.001; 364/264.300; 364/264.500; 364/267.400; 364/267.700;
              371/016.200; 395/500.000
NCL    NCLM:  395/500.440
       NCLS:  395/500.050; 395/500.340; 714/028.000; 714/030.000; 714/727.000
IC     [5]
       ICM: G06F015-20
EXF    364/578; 364/579; 364/580; 371/16.1; 371/16.2; 371/22.3; 395/500


=> d ab 31


L6   ANSWER 31 OF 32  USPATFULL
AB     The present invention concerns an integrated circuit (1) comprising a
       standard cell (4), an application cell (2) and a test cell (3)
       designed in particular to store or to modify from outside the
integrated
       circuit the value of communication signals passing between the standard
       cell and the application cell. The standard cell
     executing instructions provided on an instruction bus (3B4) by a
       program memory located in the application cell in response to
       an instruction address carried by an instruction address bus (3A4), the
       conductors of these buses constituting communication links. The
       integrated circuit further includes a branching circuit for replacing
at
       least one erroneous instruction from the program memory with a
       replacement instruction previously stored in the integrated circuit in
       response to a predetermined state of the communication links.


=> d 31


L6   ANSWER 31 OF 32  USPATFULL
AN     94:38538  USPATFULL
TI     Integrated circuit including a test cell for efficiently testing the
       accuracy of communication signals between a standard cell and an
       application cell
IN     Dartois, Luc, Colombes, France
       Dulongpont, Jacques, Pontoise, France
       Reusens, Peter, Laarne, Belgium
```

```
PA     Alcatel Radio    phone, Paris, France (non-U.S [illegible]rporation)
PI     US 5309444  1    503
AI     US 1991-739544  19910802 (7)
PRAI   FR 1990-9978        19900803
DT     Utility
LN.CNT 949
INCL   INCLM: 371/010.200
NCL    NCLM:  714/710.000
IC     [5]
       ICM: G06F011-00
EXF    371/10.2; 371/10.1; 371/8.1; 371/24; 371/21.2; 365/200; 324/73.1;
       324/158R

=> d ab 32

L6   ANSWER 32 OF 32  USPATFULL
AB     For use in an information processing system 10, the system including a
       system bus 16 having a system address bus 16a and a system data bus 16b
       and at least two data processors 12 and 14 coupled to the system bus,
       emulation apparatus for enabling a first one of the data processors to
       execute, in conjunction with a second one of the data processors, a
       program requiring access to predetermined address locations associated
       with a specific type of device, typically an I/O device. The specific
       type of device is either not resident within the system or is resident
       at different address locations. The emulation apparatus includes
       circuitry 30 for detecting an occurrence of an access cycle by the
first
       data processor to the predetermined address location, circuitry 50 for
       halting the first data processor before completion of the access cycle
       and circuitry 50 for notifying the second data processor that the first
       data processor is halted. The emulation apparatus further includes
       circuitry 24a for indicating to the second data processor a value of
the
       predetermined address location being accessed and a type of access to
       the predetermined address location. The second data processor includes
       circuitry for interpreting the address value and type of access, for
       accessing a corresponding address location having a same type of
       specific device or a corresponding type of device and for causing the
       first data processor to be released to complete the access cycle.

=> d 32

L6   ANSWER 32 OF 32  USPATFULL
AN     92:17564  USPATFULL
TI     Information processing system emulation apparatus and method
IN     Morss, Stephen, Somerville, MA, United States
       Dreyfus, Boris, Lexington, MA, United States
PA     Wang Laboratories, Inc., Lowell, MA, United States (U.S. corporation)
PI     US 5093776  19920303
AI     US 1989-367297  19890615 (7)
DT     Utility
LN.CNT 738
INCL   INCLM: 395/500.000
       INCLS: 364/232.300; 364/927.810; 364/DIG.001
NCL    NCLM:  395/500.460
       NCLS:  395/500.450
IC     [5]
       ICM: G06F009-00
EXF    364/927.81; 364/578; 364/232.3
```

```
=>
Connection closed by remote host
```

WEST

Help Logout Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Terms	Documents
L7 and ((emulat$ adj4 object) same (operating adj system))	0

Database:

US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L11

Refine Search

Recall Text Clear

Search History

DATE: Tuesday, May 20, 2003 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=OR		
L11	L7 and ((emulat$ adj4 object) same (operating adj system))	0	L11
L10	L7 and (emulat$ adj4 object)	2	L10
L9	L7 and (emulat$ same object)	54	L9
L8	l7 and (emulation adj object)	0	L8
L7	((703/24)!.CCLS. |(705/7 |705/16 |705/21 |705/23 |705/1)!.CCLS. |(714/38)!.CCLS. |(or/)!.CCLS.)	2177	L7
L6	L3 and (emulation adj object)	0	L6
L5	L3 and ((operating adj system) same (direct$ or connect$ or link$ or interfac$))	2	L5
L4	L3 and (operating adj system)	2	L4
L3	(5088033 or 5812668)[pn]	2	L3
L2	L1 and ratio	1	L2
L1	(6092048 or 5907490 or 5765140 or 5301320)[pn]	4	L1

END OF SEARCH HISTORY